



Research Repository UCD

Title	The Impact of the English Language on the Academic Performance of Non-Native English-Speaking Students in CS1 Courses
Authors(s)	Alaofi, Suad Awad
Publication date	2025
Publication information	Alaofi, Suad Awad. "The Impact of the English Language on the Academic Performance of Non-Native English-Speaking Students in CS1 Courses." University College Dublin. School of Computer Science, 2025.
Publisher	University College Dublin. School of Computer Science
Item record/more information	http://hdl.handle.net/10197/30051

Downloaded 2026-10-24 21:25:23

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information



The Impact of the English Language on the Academic Performance of Non-Native English-Speaking Students in CS1 Courses

by

Suad Awad Alaofi

Student Number: 18208628

This thesis is submitted to University College Dublin in fulfilment of
the requirements for the degree of Doctor of Philosophy

School of Computer Science

Head of School: Assoc. Prof. Neil Hurley

Principal Supervisor: Assistant Professor Seán Russell

Doctoral Studies Panel Members:

Assis. Prof. Vivek Nallur

Assis. Prof. Brett Becker

May 2025

Statement of Authorship:

I hereby certify that the submitted work is my own work, and was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Signature: *Suad Awad Alaofi*

Date: *May 01, 2025*

ABSTRACT

The challenges that non-native English-speaking students may face in English-based programming courses have been mentioned in several studies. Many of these studies have considered proficiency in English as a principal factor contributing to overall student performance, but very few have explored the impact of other English-related factors. Additionally, these studies were conducted in varied classes at different stages of the students' education. Few of them focused on introductory programming courses or CS1, where students have their first formal experience with programming.

The main focus of this research is to investigate the impact of three English-related factors on the academic performance of students in CS1 courses. The factors investigated are general English proficiency, knowledge of computer terminology, and foreign language classroom anxiety. In addition, this study explores students' perspectives on the difficulties they face while studying CS1 courses in English and using English-based programming languages. Finally, the effect of a peer-review-based online learning activity on students' knowledge of course-specific computer terminology was examined. This activity was designed to improve students' knowledge of, and confidence in, computer terminology.

This research applies the multiple-case study methodology. The choice of this methodology aimed to overcome the limitations that may affect the generalisability of the findings when only a single case study is conducted. CS1 students from two universities in different non-native English-speaking countries were invited to participate in this research. Over three academic years, more than 300 students from two different language backgrounds participated in the study. Quantitative and qualitative data were collected using various research instruments, including a standard English test, computer terminology test, and an adjusted foreign language classroom anxiety scale, as well as other surveys. Finally, the peer-based learning activity was evaluated using a quasi-experimental approach.

The investigations within this study identified several interesting correlations. Students' knowledge of computer terminology was found to have a significant influence on their academic performance in CS1 courses. General English proficiency also showed a low but statistically significant correlation with academic performance in the same courses. A weak negative correlation was identified between foreign language classroom anxiety levels and academic performance, though it was statistically significant in only one of the case studies.

Based on the influence of computer terminology knowledge, an online managed and peer-reviewed translation activity was developed to help non-native English-speaking students improve their knowledge of such terminology. Students who participated in this activity performed, on average, 8% better on the computer terminology test than students in the same class from previous academic years, when the activity was unavailable.

The exploration of students' perspectives highlighted the main difficulties they face in CS1 courses regarding the use of English as a medium of instruction and studying English-based programming languages. Asking and answering questions in English, understanding programming concepts when explained in English, and understanding computer terminology were found to be the main challenges faced by non-native English-speaking students in English medium instruction classes. With respect to English-based languages and tools, dealing with error messages, using the English interface of the coding software, and understanding keywords represented the top challenges. Finally, the majority of the surveyed students expressed a desire to use a combination of English and their native language in CS1 courses.

This research is the first to conduct an in-depth investigation into the impact of English on the academic performance of non-native English-speaking students in English-medium instruction CS1 courses. The research makes several contributions, including a large-scale literature review, the development and validation of the first computer terminology test and word list for CS1 courses, and the creation of an online learning activity to improve students' knowledge of computer terminology. It also investigates and presents the correlation between students' academic performance in CS1 courses and their English proficiency, knowledge of computer terminology, and foreign language classroom anxiety using a large sample of participants. Additionally, non-native English-speaking students' perspectives on studying CS1 through English medium instruction and studying English-based programming languages are included. Finally, this study highlights important directions and research questions to guide future research in this area.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to Almighty God for granting me numerous blessings, knowledge, and opportunities. These blessings have enabled me to accomplish many things in my life, including this thesis.

God's blessings came to me in many ways, the best of which is having amazing people in my academic and personal life. With heartfelt appreciation, I would like to take this opportunity to express my gratitude to the extraordinary people who have helped me along the way during my PhD studies.

I am grateful to Dr. Seán Russell, my principal supervisor, for his guidance and feedback throughout my research. Despite the challenges posed by the COVID-19 pandemic during my study period, Dr. Russell was always available to answer my questions and provide valuable advice. He constantly motivated me to do more but never put any undue pressure on me. He always manages to put a positive spin on negative feedback. His positivity was a great help with the stress involved in completing this thesis. Without his support, this work would not have been possible, and my study journey would have been much harder.

I would also like to extend my gratitude to my doctoral studies panel members, Dr. Vivek Nallur and Dr. Brett Becker, who provided me with very helpful feedback and advice. A special thanks goes to Dr. Becker for inspiring me with the main idea of this research. I would also like to thank everyone at University College Dublin who helped me during my studies.

My beloved parents, my first supporters, I am grateful for the life they gave me despite all the challenges they had to go through. I would not have achieved what I have accomplished without their unconditional love, support, and encouragement. No words can express my appreciation for everything they have done and continue to do for me.

I am grateful to my brother Mazen for always believing in me and supporting me whenever I needed it. He is the kind of brother every girl should have.

A special thanks to my sisters Azizah and Kholoud, whom I feel incredibly fortunate to have in my life. They lifted me up when I felt down, celebrated my achievements as if they were their own, and were always there for me. My only wish is to have them around for the rest of my life.

During my stay in Dublin, I had the pleasure of meeting some wonderful people, for whom I am extremely grateful. Although doing my PhD was an incredibly stressful time, the time I spent with them made the experience much more enjoyable and exciting. The memories we made together will stay with me for life.

I was once told to always be grateful for others but not forget to be thankful to myself. The PhD is a highly individualised educational stage, which makes it more of a personal journey. For me, the COVID-19 lockdowns made a large part of this journey even more isolated, especially because I lived away from my family. Despite all the negativity and uncertainty that the world was going through at that time, I managed to remain positive, focused, and productive. Therefore, I would like to take the opportunity to thank myself for being resilient, using that time to improve my academic skills, and making significant progress in my research. I am grateful to myself for all the lessons I have learned from the ups and downs I had to go through during this journey. The skills and knowledge I have gained over the past five years go beyond academic and scientific knowledge. The life skills I managed to acquire while pursuing my PhD are immeasurable.

Finally, I would like to thank Shaqra University for the generous scholarship they awarded me to complete my PhD studies. I would also like to thank the Saudi Cultural Bureau in Dublin for their collaboration in facilitating this scholarship.

*To the journey that shaped me, the challenges that strengthened me, and the
people who stood by me.*

CONTENTS

Abstract	iii
Acknowledgements	vi
List of Publications	xiv
1 Introduction	1
1.1 Research Background	1
1.2 Problem Statement and Motivation	4
1.3 Aims and Objectives	6
1.4 Research Questions	7
1.5 Research Hypotheses	8
1.6 Thesis Structure	8
2 Literature Review	11
2.1 Introduction	11
2.2 The Scope and the Search Strategy	11
2.3 Academic Performance in Programming and English Proficiency	13
2.4 The Challenges Encountered by NNES Students in EMI Programming Classes	14
2.4.1 English Language	15
2.4.2 English Computer Terminologies	16
2.4.3 Reading and Writing Code	19
2.4.4 Lack of Study Materials in Students' Native Language	21
2.4.5 English Integrated Development Environment (IDE)	22
2.4.6 Foreign Language Anxiety	23
2.4.7 Cognitive Load	26
2.5 Native Vs. Non-Native English Speaking Students in Programming Courses	28
2.6 English Medium Instruction (EMI) in Programming Courses	32
2.7 Factors that Correlate with Students' Achievement in Programming Courses	36
2.8 Support Provided to NNES Students in EMI CS1 Courses	38
2.8.1 Curricular and Pedagogical Approaches	38
2.8.2 Specially Designed Exams	45

2.8.3	Non-English-based Programming Languages	46
2.8.4	Translation of Code and Error Messages	49
2.8.5	Computer Terminology Teaching	50
2.9	Literature Context, Gaps and Weaknesses	54
2.10	Conclusion	58
3	Research Methodology, Methods, and Design	59
3.1	Introduction	59
3.2	Theoretical Framework	59
3.3	Research Methodology	61
3.4	Research Methods and Design	63
3.4.1	Phase One: Research Instrument Development and Data Col- lection	64
3.4.2	Phase Two: Data Analysis	70
3.4.3	Phase Three: Cross-Case Analysis and Hypotheses Testing . . .	74
3.5	Participants and Sample Size	75
3.6	Ethical Considerations	76
3.7	Conclusion	77
4	Computer Terminology Test: Development and Validation	79
4.1	Introduction	79
4.2	CS1 Word List	80
4.2.1	Background	80
4.2.2	Methods and Tools	82
4.2.3	Results	83
4.3	Computer Terminology Test (CTT)	87
4.3.1	Development Process	87
4.3.2	Test Validation Background	88
4.3.3	Validation Methods and Tools	90
4.3.4	Validation Results	90
4.3.5	Validation Summary and Discussion	95
4.4	Conclusion	96
5	Data Analysis and Results	98
5.1	Introduction	98
5.2	Data Analysis Procedure	98
5.3	Case Study One (China)	99
5.3.1	The Participants	99
5.3.2	The Students' Perspectives	101
5.3.3	Correlation Analysis Results	105
5.3.4	Multiple Linear Regression Analysis Results	107
5.4	Case Study Two (Saudi Arabia)	109
5.4.1	The Participants	109
5.4.2	The Students' Perspectives	110
5.4.3	Correlation Analysis Results	114
5.4.4	Multiple Linear Regression Analysis Results	116
5.5	Cross-Case Analysis	117

5.5.1	Participants' Background	117
5.5.2	The Students' Perspectives	119
5.5.3	Correlation Results Comparison	121
5.5.4	Multiple Linear Regression Results Comparison	126
5.6	Research Hypotheses Testing	127
5.7	Conclusion	129
6	Managed and Peer-Reviewed Translation Activity for Computer Terminology	130
6.1	Introduction and Motivation	130
6.2	The Activity Design	132
6.3	The Tool Design	134
6.4	Experiment Design and Implementation	136
6.4.1	Approach	136
6.4.2	Participants	137
6.4.3	Selected Terms and Timing	137
6.4.4	Students' Feedback Survey Design	138
6.5	Experiment Results	139
6.5.1	The CTT Results	139
6.5.2	Students' Feedback Result	141
6.6	Conclusion	143
7	Discussion	145
7.1	Introduction	145
7.2	Research Question One and Hypotheses Evaluation	145
7.3	Research Question Two	154
7.4	Research Question Three	156
7.5	Conceptual Model for the Research Findings	162
8	Conclusion	167
8.1	Research Overview	167
8.2	Significance and Contributions	168
8.3	Implications and Recommendations	171
8.4	Limitations and Future Work	173
8.4.1	Theoretical and Empirical Considerations	173
8.4.2	Methodological Considerations	175
8.4.3	Ethical Considerations	177
8.4.4	Future Research Directions	177
8.5	Concluding Remarks	180
	Bibliography	181
	Appendices	213
	Appendix A: Background Survey	215
	Appendix B: Computer Terminology Test	222
	Appendix C: Adjusted Foreign Language Classroom Anxiety Scale	227
	Appendix D: Literature Review Analysis for Section 2.9	230
	Appendix E: Item Statistics Results	236
	Appendix F: Distractor Analysis Results	237

Appendix G: Published Paper: The Influence of Foreign Language Classroom Anxiety on Academic Performance in English-Based CS1 Courses. . . .	238
---	-----

LIST OF TABLES

3.1	Summary of research instruments.	69
3.2	Summary of the statistical analysis techniques.	73
3.3	Number of participants in the multiple case study	76
4.1	CS1 Word List	84
4.2	Matching percentage between the CS1 word list and previous word lists	85
4.3	The 30 most important terms in the CS1 word list	87
4.4	Summary of Item Statistics	91
4.5	Selected Item Statistics (The full result is available in Appendix E) . .	93
4.6	Summary of Person Statistics	95
5.1	Number of participants - Case Study One (China)	100
5.2	Difficulties with using English in CS1 class - Case Study One (China)	102
5.3	Difficulties with English-based programming languages - Case Study One (China)	104
5.4	Correlation Results - Case Study One (China)	107
5.5	Model Summary ^b - Case Study One (China)	107
5.6	ANOVA ^a - Case Study One (China)	108
5.7	Multiple linear regression coefficients ^a - Case Study One (China) . . .	108
5.8	Number of participants - Case Study Two (Saudi Arabia)	110
5.9	Difficulties using English in CS1 class - Case Study Two (Saudi Arabia) . . .	111
5.10	Correlation Results - Case Study Two (Saudi Arabia)	115
5.11	Model Summary ^b - Case Study Two (Saudi Arabia)	116
5.12	ANOVA ^a - Case Study Two (Saudi Arabia)	116
5.13	Multiple linear regression coefficients ^a - Case Study Two (Saudi Arabia)	117
5.14	A comparison of participants' backgrounds	118
5.15	Comparing Students' perspectives.	122
5.16	SET Correlation Comparison	123
5.17	CTT Correlation Comparison	124
5.18	FLCAS Correlation Comparison	125
5.19	Linear Regression Comparison	126
5.20	Hypotheses Testing Results	128
6.1	The computer terminology used in the translation activity	138

6.2	Descriptive summary comparing the control and experimental groups' performance in CTT	140
6.3	What students like about the translation activity	142
6.4	What students dislike about the translation activity	143

LIST OF FIGURES

1.1	"Hello, World!" code written in Alif, an Arabic-based programming language.	2
2.1	The number of research published from 2001 until the end of 2023 that included NNES Programming or CS Students in the participants.	55
3.1	Data Collection Timeline	70
3.2	Data preparation and instruments validation process	71
4.1	The steps of generating the CS1 word list	83
4.2	The overlap between previous word lists and the CS1 word list	85
4.3	Person-Item Map	92
4.4	Between-groups differential test functioning (DTF)	94
5.1	Difficulties with using English in CS1 class - Case Study One (China)	101
5.2	Difficulties with using English-based programming languages - Case Study One (China)	103
5.3	Students' perspectives about using English in CS1 course - Case Study One (China)	105
5.4	Scatterplots matrix - Case Study One (China)	106
5.5	Difficulties with using English in CS1 class - Case Study Two (Saudi Arabia)	110
5.6	Difficulties with using English-based programming languages - Case Study Two (Saudi Arabia)	112
5.7	Students' perspectives about using English in CS1 course - Case Study Two (Saudi Arabia)	113
5.8	Scatterplots matrix - Case Study Two (Saudi Arabia)	114
6.1	Translation activity process	134
6.2	Students' performance in CTT	140
6.3	Students' feedback on the translation activity	142
7.1	Conceptual model	162

LIST OF PUBLICATIONS

- Alaofi, S. **'The Impact of English Language on Non-Native English Speaking Students' Performance in Programming Class.'** in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, poster, 2020, 2 pages. Available at: <https://doi.org/10.1145/3341525.3394008>
- Alaofi, S. and Russell, S. **'Computer Terminology Test for Non-native English Speaking CS1 Students.'** in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE)*, poster, 2021, 1 page. Available at: <https://doi.org/10.1145/3408877.3439593>
- Alaofi, S. and Russell, S. **'A Validated Computer Terminology Test for Predicting Non-native English-speaking CS1 Students' Academic Performance.'** in *Proceedings of the Australasian Computing Education Conference (ACE)*, article, 2022, 10 pages. Available at: <https://doi.org/10.1145/3511861.3511876>
- Alaofi, S., and Russell, S. **'The Influence of Foreign Language Classroom Anxiety on Academic Performance in English-based CS1 Courses.'** in *Proceedings of the United Kingdom & Ireland Computing Education Research Conference (UKICER)*, article, 2022, 7 pages. Available at: <https://doi.org/10.1145/3555009.3555020>
- Russell, S., Alaofi, S., and Alshaigy, B. **'Translate Together: Managed Translation and Peer-Review.'** in *Proceedings of the Innovation and Technology in Computer Science Education Conference (ITiCSE)*, short paper, 2023, 2 pages. Available at: <https://doi.org/10.1145/3587103.3594160>
- Alaofi, S., and Russell, S. **'The Use of English Language to Teach CS1 to Non-Native English Speakers: Students Perspective.'** in *Proceedings of the ACM Global Computing Education Conference (CompEd)*, article, 2023, 7 pages. Available at: <https://doi.org/10.1145/3576882.3617931>
- Alaofi, S., and Russell, S. **'Improving Knowledge of CS1 Terminology Through a Peer Reviewed Translation Activity: Results and Feedback'** in *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE)*, article, 2024, 7 pages. Available at: <https://doi.org/10.1145/3626252.3630876>

INTRODUCTION

1.1 Research Background

A student's first programming course, commonly referred to as CS1, is one of the core modules in the computer science (CS) major [1]. In this course, students learn how to apply problem-solving and computational thinking skills to solve problems using a programming language. Tracing, explaining, writing, understanding and debugging simple code are the most important programming skills students develop in such courses [2]. This involves learning a programming language syntax and semantics, as well as understanding how computers interpret code to perform a programming task.

Programming courses have been considered a difficult subject for various reasons and often have high dropout rates [3]. Students in programming courses face different challenges related to syntactic knowledge, conceptual knowledge, and strategic knowledge, as well as various misconceptions about the course [4]. However, students' previous academic knowledge seems to play a major role in their ability to master programming. Several studies have attempted to identify factors that impact or predict students' academic performance in CS1 (e.g. [5, 6, 7, 8, 9]). Most of the identified factors were related to the students' previous academic performance in subjects like math, science, English, and of course, prior programming experience.

In the case of non-native English-speaking (NNES) students, English proficiency levels would be expected to have a higher impact on their programming performance than on that of native English-speaking (NES) students. Nevertheless, there is an ongoing debate among researchers regarding the extent to which this factor can impact NNES students' academic performance in such courses. It has been argued that if students have a low level of English competency, they might face challenges in learning programming languages that are English-based, which is the case for most popular pro-

programming languages [10]. It will be even more challenging if English is used as the medium of instruction in those courses [11].

To better understand this issue, let us put ourselves in the shoes of someone trying to learn programming for the first time in a language that is different from their native language. Figure 1.1 shows an example of a code written in *Alif* which is an Arabic-based programming language [12]. You might think that your competency level in Arabic would greatly influence what you see in those lines of code, which is true to some extent. Nothing will make sense, including the letters and the direction of writing unless you have at least some knowledge of Arabic language basics. This is exactly the same for NNES students in CS1 courses, especially if the alphabet of their native language is completely different from the Roman alphabet. Their English proficiency level will determine how they look at a piece of code written in an English-based programming language such as Python. NNES students must be familiar with English — at least its alphabet, writing direction, and punctuation — to begin learning programming in any English-based programming language.

```
# ألف
نافذة رئيسية
دالة رئيسية
نص أ = "أهلا"
نص ب = "بالعالم"
أ + ب = أ
رسالة (أ)
نهاية دالة
نهاية نافذة
```

Figure 1.1: "Hello, World!" code written in Alif, an Arabic-based programming language.

Most widely used programming languages strongly rely on English, including keywords, syntax, symbols, available learning materials, and coding software [10, 13, 14]. One of the main reasons for designing high-level programming languages to resemble natural language is to make them easier to learn and use [15]. Moreover, high-level programming languages aim to allow programmers to focus more on problem-solving and logic rather than hardware-related details [16]. Some previous studies concluded that teaching programming using a programming language based on the learners' native language helped them to understand programming concepts faster

and better [17, 13, 18]. Accordingly, when it comes to popular English-like programming languages, this advantage will only be applicable to NES learners, while NNES learners will be at a disadvantage.

Several attempts have been made to create programming languages based on different natural languages, but they have not been successful in the tech industry. For this reason, software companies continued to develop English-based programming languages, even in non-English-speaking countries, and Python, which was developed in the Netherlands, is a well-known example [19]. The available non-English-based programming languages are very basic and can only be used for educational or research purposes. Therefore, learners who wish to pursue a CS major or consider programming as a profession will ultimately need to master at least one popular programming language. NNES programming learners, particularly those with no or low English proficiency, usually face a real challenge when learning these English-like programming languages.

Let us go back to the code example in Figure 1.1. Now, imagine someone started to explain this piece of code to you in Arabic. Would you be able to follow them? Again, your level of Arabic proficiency might be the first factor to consider. A growing number of higher education institutions worldwide are implementing English Medium Instruction (EMI) [20], particularly in science disciplines such as CS. NNES students applying for CS majors where EMI is used are usually required to possess a certain level of English proficiency to be offered a place. However, this requirement does not guarantee that their level of English will not influence their academic performance in such courses. This can be concluded from the results of some research, where NNES students were found to be at a disadvantage when compared to NES students in EMI courses (e.g. [21, 22, 23]).

Therefore, this research focuses on investigating the impact of different factors related to English and their influence on NNES students' academic performance in CS1 courses. Over 300 first-year university students from different language backgrounds, institutions, and countries participated in this study. More than 15 CS1 courses were included in this investigation, and data were collected over three academic years, making it one of the largest studies in scale to consider the challenges facing NNES students in such courses.

1.2 Problem Statement and Motivation

To continue in the CS major, undergraduate students are generally required to pass the introductory programming course, commonly referred to as CS1, offered in their first year of the programme. However, the dropout rate in this particular course is concerning, ranging between 30% and 40% [24, 25]. This number of students dropping out can be higher in some institutions. For example, a study at one of the Saudi universities revealed that 50% of the students dropped out of their CS1 course [26]. This is not a local problem related to a specific institution or geographical location, but rather a more global issue [27]. The impact of difficulties students face in their first year in CS, particularly a programming course, sometimes results in dropping the major altogether [28]. Many factors might influence the students' decision to drop out of CS1 courses, one of them being the difficulties arising from a low proficiency in the English language. Horton and Craig [22], in their study, noticed that students who are not fluent in English are more likely to fail or drop the programming course, especially when much reading and writing in English is required. Some students even dropped the course despite having a chance to pass. Therefore, it is impossible to estimate the number of talented students who have been excluded from the CS field as a result of language barriers.

Many studies have been conducted over the years to identify and address challenges affecting CS students' learning in programming classes. Although NNES students were not the main focus of most of these studies, some did identify low English language proficiency as one of the challenges programming students face [29, 30, 31]. Despite the disagreement among researchers on the correlation level between English proficiency and students' academic performance in programming courses, none of them denied the negative influence that low English competency may have on students' achievements.

Another influencing factor that has not been widely discussed in CS is students' knowledge of domain-specific English terminology and its relationship with their academic performance. In programming, the high density of English computer terminology in learning materials has shown a negative influence on students' academic achievements [32]. Another recent study linked students' academic performance in a non-programming introductory course to CS and their level of knowledge of computer terminology [33]. In a closely related field, research on algebra courses has shown that NNES students need a high proficiency in everyday English as well as specialised mathematical vocabulary to fully comprehend the course learning material [34]. Yet, research on the impact of students' knowledge of CS terminology on their academic

performance in CS1 courses, as well as how to measure or improve their knowledge of such terminology, is still very limited.

In addition to the mastery of general English and specific-field terminology, programming students are required to learn a programming language, which some researchers consider as difficult as learning a natural language [35, 36]. Most popular programming languages taught in CS1 courses are English-based, meaning they consist of keywords derived from the English language [13]. Even some programming language symbols are similar to punctuation systems used in English, which is difficult for some NNES students to follow [37]. The syntax of some of these programming languages is also heavily influenced by English [14]. It is also important to note that most available and updated coding tools, documentation, learning materials, and popular discussion forums are in English [10]. The lack of high-quality translations of learning materials and CS terminology in different languages is one of the issues that NNES students need to deal with [38]. Even though some research has explored the challenges faced by novice NNES programmers, limited studies have focused on students enrolled in formal CS1 courses at the university level and the impact of these challenges on their academic performance. Understanding these students' perspectives is crucial, particularly for institutions seeking to improve the students' learning experience and ensure their continued success in the programme.

Another dimension of English-related difficulties is the cognitive overload that NNES students might experience in EMI programming courses. NNES students in those classes are required to learn new programming concepts and a new programming language in a foreign language, which could result in a very high cognitive load [39]. NNES students' progress in programming courses could be highly influenced by the cognitive overload they might experience in class [40]. The negative impact of cognitive overload was noticed in NNES students' results in final exams in some CS subjects [41, 21]. Besides the cognitive overload, NNES students in different majors were found to experience different levels of foreign language anxiety in EMI courses [42, 43, 44, 45]. This factor was found to negatively influence students' academic achievements in non-CS majors [46]. However, there are no available studies investigating its impact on CS students' academic performance, particularly, in CS1 courses.

Traditionally, CS1 courses are offered to students in their first year at university in CS or closely related majors, such as IT and engineering. As a growing number of universities adopting EMI worldwide [47], these courses may be among the first EMI courses these students have ever taken. A considerable number of NNES students register every year for such courses in many countries, sometimes coming from abroad

specifically for study. Many universities rely heavily on international students as a source of funding, as well as for the general economic benefits they bring to the host countries. In the UK, for example, a fifth of universities' income comes from international students [48], and figures show that they boosted the UK economy by £41.9 billion in 2020/2021 alone [49]. As previously mentioned, the challenges that students encounter during their first programming course can lead them to drop out of their major [28]. This could cause a financial loss for the educational institution, and it would be a greater loss if the students are international, as they typically pay higher tuition fees. Therefore, understanding the difficulties these students face and working on helping them overcome these challenges will have great benefits for all parties.

Finally, the shortage of skilled programmers is a global issue, and the demand is increasing year on year. In 2021, the shortage of qualified software developers and tech talent was estimated to be 40 million globally, and this number is expected to reach 85.2 million by 2030 [50, 51]. This can cause a huge loss for the economy as the unrealised annual revenues because of this shortage could be about \$ 8.5 trillion by 2030 [51]. One of the suggested ways to tackle this problem is to widen the source of talent and engage more diverse groups in the tech industry, starting with education. CS1 courses might be the first real experience for many students, and future programmers, to learn one of the high-level programming languages demanded by large companies in the tech industry. Therefore, lowering barriers that NNES students might face in those courses might ensure their continuing in the major which is a crucial step to address the global shortage of developers as it will bring more diversity to the field.

1.3 Aims and Objectives

This research has three main aims, the first is to investigate the impact of the English language on NNES students' academic performance in EMI CS1 courses. As the impact of English could have an unlimited number of related factors, this study focuses on three factors: level of general English proficiency, level of English computer terminology knowledge, and foreign language classroom anxiety level. The second aim is to identify effective methods to help students enhance their knowledge of computer terminology through the design, implementation, and evaluation of an online learning activity. The third aim is to explore NNES students' perspectives on studying in an EMI environment and using English-based programming languages in their first formal programming course.

The research will be guided by objectives to ensure it remains focused on the research aims and avoids ambiguity. These objectives include:

1. To investigate the correlation between English language proficiency level and NNES students' academic performance in the CS1 course.
2. To examine the correlation between NNES students' knowledge of computer terminology and their academic performance in the CS1 course.
3. To investigate the relationship between NNES students' academic performance in English-based CS1 courses and the level of foreign language anxiety.
4. Examine to what extent online managed and peer-reviewed translation activity for computer terminology can improve NNES students' knowledge of terminology related to the CS1 course they are attending.
5. To explore the NNES students' perspectives on studying CS1 courses in EMI and using English-based programming languages.

1.4 Research Questions

The study seeks to answer three main questions in order to achieve the research aims and objectives. Those questions have been formatted to be focused, clear, and within the limit of the research scope. Following are the three research questions along with sub-questions:

RQ 1: What is the influence of the following three factors on NNES students' academic performance in the CS1 course:

1. English proficiency level,
2. Knowledge of computer terminology,
3. Level of foreign language classroom anxiety.

RQ 2: To what extent can online managed and peer-reviewed translation activity for computer terminology improve students' knowledge of terminology related to the CS1 course they are attending?

RQ 3: What are the perspectives of NNES students on using the English language in CS1 courses?

- RQ3.1** What difficulties do NNES students perceive in EMI CS1 courses?
- RQ3.2** What difficulties do NNES students perceive when studying English-based programming languages?
- RQ3.3** To what extent do NNES students consider the English language a barrier to learning programming?
- RQ3.4** What language(s) do students prefer to use in CS1 class?

1.5 Research Hypotheses

In light of the extensive literature review, three main English-related factors that may impact NNES students' academic performance in CS1 courses were identified. As a result, a set of hypotheses were formulated to be tested and used to address the first main research question.

- **Hypothesis 1:** *The level of English language proficiency impacts NNES students' academic performance in CS1 courses.*
- **Hypothesis 2:** *The NNES students' level of knowledge in English computer terminology impacts their academic performance in CS1 courses.*
- **Hypothesis 3:** *Foreign language classroom anxiety level impacts NNES students' academic performance in CS1 courses.*

1.6 Thesis Structure

- **Chapter 1: Introduction**

This chapter aims to provide an overview of the research topic, its aim, objectives, questions, and hypotheses.

- **Chapter 2: Literature Review** This chapter reviews related work in the field, covering all aspects investigated in this research. It explains the generation of the research hypotheses and justifies the research questions. Given the limited research on NNES students in EMI CS1 courses, the review expands to include studies on NNES students in related subjects, other science majors, and various educational levels. It also includes recent studies not covered in the published work, particularly in the related work sections.

- **Chapter 3: Research Methodology, Methods, and Design**

This chapter presents details on the methodology, methods, and design of the research, along with the justification for the approaches and instruments used. It also outlines the data collection and analysis process. Finally, information about the participants, sample size, and the ethical considerations involved in the research are provided. Most of our published work relies on the methods outlined in this chapter, so many sections of this chapter were already included in introducing the methods adopted in our published work.

- **Chapter 4: Computer Terminology Test: Development and Validation**

This chapter provides more details on the development and design of one of the main research instruments, the Computer Terminology Test. It discusses all stages of building the test and the detailed validation process. A separate chapter has been dedicated to this instrument because no existing tools were suitable. There was a need to develop and validate a specific test for this research, as well as for future research and educational purposes. Most of the material in this chapter has already been presented and published in a conference poster and another conference paper. The related publications are:

- Poster: *'Computer Terminology Test for Non-native English Speaking CS1 Students.'* In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE), 2021.
- Research Paper: *'A Validated Computer Terminology Test for Predicting Non-native English-speaking CS1 Students' Academic Performance.'* In Proceedings of the Australasian Computing Education Conference (ACE), 2022.

- **Chapter 5: Data Analysis and Results**

This chapter presents data analysis and results of all data collected over two academic years. As this research included two case studies, the results of each case study are presented separately followed by cross-case analysis. Data are provided in the form of figures, charts or tables as well as written explanations. Some of the results presented in this chapter have already been published in various conferences. The related published work includes:

- Research Paper: *'The Influence of Foreign Language Classroom Anxiety on Academic Performance in English-based CS1 Courses.'* In Proceedings of the United Kingdom & Ireland Computing Education Research Conference (UKICER), 2022.

- Research Paper: *'The Use of English Language to Teach CS1 to Non-Native English Speakers: Students Perspective.'* In Proceedings of the ACM Global Computing Education Conference (CompEd), 2023.

- **Chapter 6: Managed and Peer-reviewed Translation Activity for Computer Terminology**

In this chapter, the design and implementation of a managed and peer-reviewed terminology translation activity is presented. The approach, participants and evaluation of the new activity in a CS1 course are discussed in detail. Moreover, the results of the implementation of this activity using the specially designed online tool are presented in this chapter. The work included in this chapter is presented in the following published papers:

- Short Paper: *'Translate Together: Managed Translation and Peer-Review.'* In Proceedings of the Innovation and Technology in Computer Science Education Conference (ITiCSE), 2023.
- Research Paper: *'Improving Knowledge of CS1 Terminology Through a Peer Reviewed Translation Activity: Results and Feedback.'* In Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE), 2024. (Accepted).

- **Chapter 7: Discussion**

This chapter discusses and reflects on the research results in light of previous research in the field. It revisits the research questions one by one, presents the findings, and connects them to related available work. This chapter reevaluates the research hypotheses by considering the findings and concludes whether they should be accepted or rejected. Some parts of this chapter have been included in the published work listed in the description of the previous chapters.

- **Chapter 8: Conclusion**

This chapter briefly summarises the research, highlighting the key findings and contributions. It also shows how the findings can be used and applied to improve the programming education provided to NNES students. Finally, it clarifies the research limitations and provides suggestions for future research directions.

LITERATURE REVIEW

2.1 Introduction

This chapter explores the available literature on the impact of English and related factors on the academic performance of NNES students in EMI CS1 courses at the university level. Given the limited research directly addressing this topic, a broader scope of literature has been considered, including studies conducted at various educational levels, on courses similar to or related to programming, and in both formal and informal educational settings. Although all English-related factors were considered, this review focuses on three primary factors: English proficiency, students' knowledge of computer terminology, and foreign language classroom anxiety. Additionally, this review examines different forms of support proposed by researchers to aid NNES students in programming courses concerning their language needs.

2.2 The Scope and the Search Strategy

This section outlines the strategy employed for conducting the literature review on the thesis's primary topic. Considering the limited research directly addressing this issue, the review includes relevant literature from broader studies where the subject is mentioned as a sub-theme. This review covers studies on programming or closely related subjects that considered the native language of students, even when NNES students were not the main focus. The literature spans from 2001 to 2023, although some earlier studies with relevant findings were included. The process used in this review was flexible and allowed for an iterative search, considering the nature of the research period of the thesis.

The purpose of the literature review was to:

- Explore existing knowledge on the topic and map out key concepts.
- Identify gaps in the literature that highlight the need for further research.
- Shape the main research questions and hypotheses.
- Help select and build the research methodology and methods.
- Provide context for interpreting and discussing results at the end of the research project.

The search was conducted across a range of databases, including, but not limited to, ACM Digital Library, IEEE Xplore, SpringerLink, ScienceDirect, and the Education Resources Information Center (ERIC). These databases were selected for their coverage of academic, technical, and education-focused publications, as well as their commitment to high-quality publication standards. Additionally, Google Scholar was used to expand the search for publications from various publishers, and relevant PhD theses and Master's dissertations were also included. However, personal blogs, self-published papers, and works lacking publication information were excluded. Only English-language publications were considered, covering participants from K-12 to university levels, as well as both formal and informal educational settings.

To ensure that all relevant literature is included, a combination of keyword searches was employed. The primary keywords were tailored to the specifics of the research topic and included: 'Non-native English speakers', 'Non-native English', 'English Learners', 'English as a Second Language', and 'native language'. Those terms were combined with one of the following terms: 'computer science', 'programming', and 'information technology'. After conducting the exploratory search, some terms were used to search for literature on specific sub-topics, including: 'English computer science terminology/ terminologies/ vocabulary', 'English medium of instruction', 'EMI', 'cognitive load/ overload', 'IDE', 'anxiety', and 'foreign language anxiety'. Boolean operators ('AND', 'OR') were used to refine the search results and identify studies that addressed the intersections of these themes.

Data from the selected articles were extracted and analysed using a structured template that included the following fields: Author(s) and Year, Title and Source, Aim, Methodology and Methods, Participants (number, age, educational level, English language proficiency level), Key Findings related to the thesis topic, and Limitations. After conducting exploratory reading, several themes were identified, and the collected publications were categorised according to these themes, with each theme covering a different sub-topic. Some publications were categorised under more than one sub-topic based

on their results. The sub-topics or themes evolved throughout the study, with some being split and others merged until the final themes were established.

2.3 Academic Performance in Programming and English Proficiency

For those learning programming, having a good command of English is either necessary, preferred, or both [52]. In a study conducted in Hong Kong, prior academic skills, including English, were found to affect students' academic performance in programming courses [53]. Similarly, Zusana E. Pudyastuti and Jozua F. Palandi [54] conducted a quantitative study to assess the English proficiency of 99 students and its effect on their programming class scores. Their findings, aligning with Mhashi and Alakeel [30], indicate that students with low English proficiency are often disadvantaged in learning programming. Other quantitative studies also identified a strong correlation between English proficiency and academic performance in computer science [55] and programming [31].

Another study identified a strong correlation between English proficiency and computer programming scores by analysing the records of 1,438 students who attended programming courses between 2007 and 2020 at an Indonesian university [56]. A similar result was found in a study that analysed 1,316 computer science alumni records, concluding that students' first-year average English course grades can significantly predict their overall grade point average (GPA) in CS [57]. The data were collected from four computer science programmes at a computer science college in Saudi Arabia.

In a software engineering course, which is relatively similar to programming, a significant but moderate correlation was found between female students' English proficiency and their academic performance [58]. The same result was also noticed in the information system program at the university level, where English competency was found to positively contribute to the academic success of the students [59].

The impact of English proficiency on learning programming has been observed at various educational levels, not just at the undergraduate level. A study on Chinese middle school students found that differences in academic performance in a programming course were directly related to their knowledge of the English language [60]. Similarly, another study identified a significant correlation between English proficiency and programming achievement at the elementary school level [61].

Contrary to popular belief, language skills, including English as a foreign language, have been found to be more important than mathematical skills for computer science and information technology students. This conclusion was drawn from a scoping review study examining the skills required for CS and IT students, with the aim of suggesting reforms to acceptance criteria [62]. In fact, the literature provides evidence that strong performance in courses such as English often leads to improved performance in programming courses [5, 63, 64, 65, 66].

However, several other studies suggest that mathematical skills or overall high school GPA are more useful than English proficiency in selecting candidates for CS programmes (e.g. [67, 68]). Moreover, many quantitative studies have found no statistical relationship between English competence and students' academic performance in CS or programming [6, 69, 64, 70, 71, 72, 73, 74, 68].

Several factors might have influenced the findings of these studies. For example, Keane and Gray [68] conducted their study in Ireland, an English-speaking country, and did not report the number of NNES students who participated. Therefore, it cannot be assumed that the findings of this study, or similar studies, apply to NNES students in general.

From the literature, it is evident that researchers have not yet reached a consensus on the strength of the correlation between English proficiency and academic performance in CS or programming courses. However, no previous research has refuted the positive influence of a good level of English on students' academic success in such programmes. Thus, the first hypothesis of this research was developed based on the literature review as follows:

Hypothesis 1: The level of English language proficiency impacts NNES students' academic performance in CS1 courses.

2.4 The Challenges Encountered by NNES Students in EMI Programming Classes

Research has found that students' success in their first CS1 course significantly impacts their decision to continue with the CS major. This course is not only crucial for setting the foundation of programming but also for shaping students' perceptions of their future in the field. NNES students face various challenges in their journey to learn programming; however, research on the difficulties they encounter in their first formal programming course is limited. The goal of this thesis is to identify and un-

derstand these challenges from different perspectives. Thus, the following sections explore the factors previously identified in research as challenges for NNES students in CS1 courses and similar courses.

2.4.1 English Language

Learning computer programming is complex, as it involves mastering various programming concepts alongside the syntax and semantics of the programming language [75]. Some research suggests that learning programming languages is akin to learning a new human language [76, 77]. This is because students must learn new terminologies and their syntax, while also developing the ability to write functional code. This challenge is particularly pronounced for NNES students, who, in many cases, must study the English language concurrently with their programming course [78, 10].

In an interesting study on the impact of culture on success in CS, Carter et al. [40] observed a number of NNES students studying programming at a US university. One of the main findings was that, although some of these students had a good level of English, they struggled to follow the lectures, ask questions, and understand them. The researchers described this case as a 'lack of comfort with English', which hindered their ability to progress in the programming class at the same level as their peers. In an empirical study, Mhashi and Alakeel [30] surveyed 143 students to determine the challenges faced by students when learning computer programming skills. The results of their study emphasised the importance of the English language for learning computer programming, as low-level knowledge of English was identified as one of the challenges these students faced.

Many schools use block-based programming, which is usually available in different natural languages, to introduce programming to new learners, especially children. However, one of the main challenges facing NNES students in their transition between these two types of programming languages is the lack of understanding of English [78].

The challenge of English proficiency is not limited to NNES students aiming to become programmers in the future but also extends to those who plan to become programming teachers. A number of NNES undergraduates in the CS teaching program expressed concerns about teaching programming while not being fluent in English [79]. The students' low proficiency in English is also a concern for those aspiring to become programming teachers.

2.4.2 English Computer Terminologies

The need for clarity and the avoidance of misunderstanding and confusion when discussing concepts within a discipline has led to the development of specialised terminologies [80]. Field-specific terminology plays a crucial role in education and scientific research; without it, students cannot receive a high-level education, and even scientists may struggle to conduct accurate research [81]. The use of English terminology has significant advantages, such as facilitating international collaboration and education, but it may pose a challenge for NNES students and potentially lead to decreased linguistic diversity in the CS field [82]. Moreover, the meaning of English terminology within a discipline may differ when the same term is used in another context or field.

The rapid technological development in the CS field has led to issues related to the inconsistent usage of certain terms [83]. In many cases, a single term can be used to refer to different concepts and the same concept can be described by different terms [83, 84]. In other words, two definitions might be used to describe the same term or two terms might be given different definitions but then used interchangeably [83, 85]. This can negatively affect communication and understanding among experts in the field and create confusion for students.

The ambiguity is not limited to unclear definitions of some terms but also extends to the use of metaphors that are no longer recognised as such by many experts in the field [86]. A metaphor can be defined as 'a description of an object or event, real or imagined, using concepts that cannot be applied to the object or event in a conventional way' [87]. An example of a popular metaphor is the '*shopping cart*' in online shopping platforms, which originally refers to a complex collection of web pages, programs, and databases [88]. Other examples of metaphors used in CS, particularly in programming, include '*string*', '*stack*', and even '*program*' and '*model*' [86]. The use of metaphors can lead to confusion for students, particularly those with low levels of English proficiency.

For CS students, particularly NNES students, it is crucial to focus on explaining this type of terminology and the distinction between its formal and informal use [86]. However, many language teachers who lack experience in other fields may not be able to explain field-specific terms, meaning that such terminology needs to be learned during the course, rather than in language classes [89]. Experts in the field have a better understanding of this terminology than language teachers, even those who teach English for specific purposes [90]. Moreover, teachers of computational thinking courses who are not CS experts also face challenges in understanding and explaining technical terms [91]. Therefore, computer terminology related to programming is better explained to students by their course instructors.

The PhD research conducted by Pal [92] examined the impact of using English versus native language as the medium of instruction in programming courses. It also compared different teaching methods, including traditional classroom teaching and screen-casting. One of the key findings is that many students, regardless of the medium of instruction or teaching method, faced difficulties in understanding computer terminology. In EMI, students struggled to comprehend all that was explained by the teacher or presented on the screen when certain computer terminology was used without sufficient explanation. Similarly, the use of translated terms in the native language instruction environment did not alleviate the issue, as some computer-specific terms in their native language were equally complex. Students expressed a preference for studying programming in EMI to improve their English but indicated a need for verbal explanations of computer terminology in their native language, particularly when first introduced.

Keen and Etzkorn [32] found in their study that students' grades in computer science class can be negatively correlated with the number of 'buzzwords' (words with computer science meaning) that appear in the lecture notes they use in class. However, this study does not clarify what buzzwords exactly are or how these words were selected. In contrast, Masrai et al. [33] investigated the correlation between scores on a general English vocabulary test (GVT) and a computer science specialist vocabulary test (CSSVT) with the academic performance of 106 Arabic-speaking students enrolled in two EMI CS courses. These targeted CS courses cover both basic and advanced CS skills, but programming is not included. The CSSVT was developed using 114 words randomly selected from the computer science word list (CSWL) published by Minshall [93]. The results revealed a significant positive correlation between the students' scores on both tests and their final course grades. The researchers suggested that further investigation is needed to confirm these results by examining students from different populations, language backgrounds, and contexts.

In a closely related study, Scott and Mason [94] interviewed 24 undergraduate students to explore the challenges NNES students face with keywords and terminology used in the Cybersecurity course they were attending. The results revealed that the majority of these students struggled with field-specific terminology, although they acknowledged its importance. This finding aligns with that of Benchechrone [95], where 83.6% of the 55 NNES Cybersecurity experts agreed on the importance of mastering field-specific terminology. Similarly, a study on the challenges faced by Vietnamese engineering technology students found that many students had difficulty comprehending field-specific content due to their struggles with English. Among many language-related

difficulties, 93.4% of the students identified English terminology as the primary challenge they faced [96].

Gawazah and Woldemariam [97] conducted a study to investigate the necessary English language proficiency requirements for undergraduate students in an information and communications technology (ICT) major. The study concluded that ICT students demonstrated a poor grasp of technical terminology, which negatively impacted their communication and writing skills. Additionally, over 50% of the surveyed students indicated that the English for Science and Technology course they were attending should place greater emphasis on technical terminology.

Ehara [98] designed a special vocabulary test for 10 English words. For each word, there were two questions: one about its regular meaning and the other about its meaning in computer science. The aim was to evaluate how challenging it is for NNES learners to understand the distinctive terms used in computer science. The researcher used a crowdsourcing service to recruit 100 Japanese participants with an intermediate level of English. The results showed that a limited number of words were particularly confusing for learners. The researcher suggested that providing NNES CS learners with a list of those words could be helpful for the learning process. The main limitation of this study is that only a one-page poster has been published, lacking many important details. Information about the participants, the test design, and the words included in the test is needed to evaluate the study's value and contribution.

Knowledge of CS terminology is crucial not only for undergraduate students but also for those who aim to pursue postgraduate studies. A survey of 40 CS Master's students revealed that field-specific terminology is the most needed language component for technical writing [99]. With the expanded use of technology in almost all other fields, the issues with CS terminology are not limited to people studying or working in the CS field. In a study on the use of CS terminology in non-computing disciplines, 75% of the interviewed educators stated that CS terminology presents a barrier to their teaching [100].

Another study also proposed that the technical terms NNES students use to explain code in their mother tongue may be useful in detecting misconceptions [101]. This was the result of a study conducted in Germany with 123 undergraduate CS1 students who were asked to explain a given code fragment in German; the survey encompassed 15 code snippets in Java [101]. The analysis of the 1800 responses showed a large variability in terms used by students, mostly in descriptions of variables and loops. In most of these responses, technical terms were correctly used, but the study suggested that analysing novice programmers' language may help identify misconceptions about programming. The researcher mentioned that 40% of the participants were from immig-

rant backgrounds; however, all the responses were collected in German. The impact of the students' native language and their proficiency level in German, the language used in the experiment, was not discussed by the authors. While the study addressed the topic of CS terminology from an interesting perspective, which is worth investigating, this thesis focuses on the impact of English CS terminology on NNES students' academic performance in CS1 courses.

Reviewing the previous literature on this topic informed the second hypothesis for the current study:

Hypothesis 2: The NNES students' level of knowledge in English computer terminology impacts their academic performance in CS1 courses.

2.4.3 Reading and Writing Code

Aside from understanding technical terminology, students also need to be able to read and understand programming code, as this enhances the problem-solving skills required for programming [102]. The influence of the English language can be seen not only in keywords but even in the syntax of most programming languages [14]. Thus, comprehending code written in an English-based programming language may be impacted by the learner's level of proficiency in the English language [103].

In a study that explored the vocalisation of code among NNES children learning programming, the majority of participants struggled with choosing between their native language and English reading [104]. Some of them used English but with their native language's pronunciation, others mixed English with their native language, while one student changed the order of the words to match their native language sentence form. They also struggled with reading keywords and pronouncing variables and symbols such as `<` and `==`.

Guo [10] surveyed 840 NNES programming learners recruited from a programming tutorial website to investigate the difficulties they face in their learning. The results showed that one of the main challenges these learners encounter is reading and writing programming code in English-based programming languages. Likewise, Qian and Lehman [60] found that keywords may be challenging for NNES students learning how to program for two possible reasons: first, not recognising the meaning of keywords, and second, difficulty with spelling.

Research results identified five main problems Chinese students struggle with when writing programming code in Python [37]. Two of these main challenges are punctuation and spelling problems. The students' low English proficiency is expected to be

the reason for the spelling mistakes, but using Chinese punctuation was an interesting finding. The researchers suggested that programming teachers should teach the spelling and meaning of programming keywords to the students to reduce spelling mistakes. Additionally, clarifying the difference between Chinese and English punctuation can help minimise errors students encounter when writing code.

The result of a study on 13 to 14-year-old students revealed that learning Scratch can help NNES students comprehend basic programming concepts better than the Logo programming language [105]. One of the main reasons for this was the use of English in Logo, which made it more difficult for students to understand programming concepts. Similarly, NNES students found it challenging to understand keywords in text-based programming languages such as Python, making their transition from block-based to text-based programming languages even more difficult [78].

Interestingly, Stefik and Siebert [106] found no significant difference between randomly generated keywords and those in well-known programming languages, such as Java, in terms of their intuitiveness to novice programmers. On the other hand, keywords in programming languages that are closer to English, such as Python, were found to be more intuitive. Since the majority of participants in Stefik and Siebert's study were native English speakers, the study was replicated by Lappi et al. [107] with NNES participants, but the results were similar. However, the participants in the replicated study were mostly proficient in English, suggesting that native English speakers and those proficient in English may have an advantage when using programming languages like Python. This does not mean that native or proficient English speakers have no issues with English programming keywords, as some keywords can be confusing if they have a different meaning in everyday English [108]. For example, a student who interprets the meaning of the programming keyword '*while*' based on its use in real life may expect the loop to terminate as soon as the test expression is satisfied [108].

Another challenge mentioned by the NNES participants in the study conducted by Guo [10] was difficulty choosing clear English identifier names when writing programming code. A study by Lawrie et al. [109] showed that using whole-word English identifiers in programming leads to better code comprehension. The study also noted that native English speakers reported difficulty reading code written by their NNES peers. A study by Elazhary [110] further corroborated this finding by emphasising that NNES programmers often have difficulty learning and writing programming code.

During an evaluation of the Hedy programming language with participants from the Netherlands, researchers were surprised to find that the students preferred to use localised keywords [111]. This was unexpected, as the Netherlands has the highest proficiency in English outside of English-speaking countries, leading the researchers to

anticipate that the students would feel comfortable using English keywords. This result supports findings from other researchers who evaluated programming languages based on the native language of the students [17, 112, 113, 114, 115].

2.4.4 Lack of Study Materials in Students' Native Language

Often, NNES learners must use study materials written in English due to the lack of resources available in their native language [38]. One of the major challenges they face is understanding English learning materials, such as textbooks, online courses, technical forums, and coding documentation [10]. This impacts their learning of programming courses, as they struggle to comprehend these materials, often encountering ambiguous explanations with no illustrations to aid understanding [71, 116, 77]. Poor and inaccurate translations of English computer terminology into the students' native language often lead to confusion [38]. For example, the translation of technical terms from English into Arabic, which is the official language of 22 countries, still falls short of the required standard [117]. Consequently, learners from minority language backgrounds are likely to face even greater challenges in accessing translated educational resources.

Moreover, NNES students face a significant challenge in understanding the content of long texts as well as writing essays and reports in English [96]. Consequently, they might need to spend extensive hours studying outside of class to fulfil the course requirements, but this does not necessarily ensure them higher scores than their NES peers who spend less time studying [118]. NNES students who attend EMI classes may also struggle to keep up with the lecture while simultaneously translating it into their native language for better understanding [40].

In programming, technical forums can be considered a type of learning resource; however, most of the popular ones are in English. In the results of the Stack Overflow user survey, a large number of users stated that the use of English is the main reason they do not participate in forum discussions [119]. The use of English can be a real barrier not only for students and beginners but also for more advanced programmers. Chinese developers, who usually have a good grasp of English, are still uncomfortable asking questions on technical forums in English and, most of the time, they face issues with translating technical terms [120].

Guzman et al. [118] explored the experiences of NNES students in four CS courses at one of the US universities. The researchers indicated that further investigation is required to understand the needs of NNES students, particularly regarding the English language barrier. Therefore, this study takes the students' perspectives into account, which has not been widely discussed in previous studies in the area.

2.4.5 English Integrated Development Environment (IDE)

The use of English in class and learning materials is not the only concern for NNES students in programming courses. Most available tools and programming platforms have only an English interface, with very limited tools providing localisation options [121, 10]. Feijóo-García et al. [122] conducted a study to explore the impact of using the English interface on NNES learners' academic performance in programming using Scratch. Their experiment included seven NES students as a control group and nine students who speak English as a second language as an experimental group. In the experiment, the tutorial and assessment were provided in English for the native-English speakers, and in Spanish, the native language of the participants, for the NNES participants. However, both groups used the programming platform, Scratch, with an English interface. The results show that there was no major statistical difference between the academic performance of NES and NNES learners. Nonetheless, the NNES respondents indicated that they would have achieved better results if the software interface had been adapted to their native language.

In a related study, Dasgupta and Hill [13] explored the connection between adapting the programming interface to the local language and the speed at which young people learn programming on Scratch. Their research included data collected from roughly 15,000 Scratch users from Portugal, Italy, Brazil, Germany, and Norway, and a total of 90,859 projects. The results showed that beginners developed programming skills faster when the platform interface was adapted to their native language compared to those who used the English interface. The main limitation of this study is the lack of information on the users' ages or level of English. Similarly, in a comparison between using Logo with an English interface and Scratch adapted to the students' native language, students found it more difficult to use or learn through the English interface [105]. The students in that study were elementary school students, and their English level was not reported.

In fact, there is evidence in the literature showing that the difficulty of compiler error messages is one of the reasons why students drop out of their computing majors [123]. It is expected that students with poor English levels might face more challenges dealing with error messages [123, 124], even though research shows no differences in the distributions of the types of error messages among NES and NNES learners [125]. This was clearly demonstrated in other studies, where students with low English competency faced challenges in using or understanding English compilation error messages, even if they were uncomplicated [110, 126]. Error messages with more jargon, 'technical terms', and acronyms are hard to read and use for novice programmers [127].

This might have a greater impact on NNES students, as only four NES students found heavy jargon to be a problem, while 43 NNES students stated this as a challenge in Denny et al. [127]’s exploratory study. This result is in line with another study investigating the challenges encountered by NES and NNES students in programming; error messages were mentioned by only 2% of NES students compared to 72% of NNES students [128, 129].

A number of researchers have worked on enhancing the readability of error messages by making them look more like natural language, resulting in positive outcomes [130, 131, 127, 132]. Another approach to helping CS1 students deal with error messages is providing them with a debugging manual [133]; however, extending these efforts to different human and programming languages still poses a significant challenge [134]. More discussion about improved error messages is provided in section 2.8.4 of this chapter.

2.4.6 Foreign Language Anxiety

University-level students have many sources of anxiety, especially first-year students [135]. A new study environment, new social life, and new subjects to study are only a few examples of these sources. For CS students, there are many causes of anxiety, including programming, computer usage, mathematics, and exams [136]. For example, increasing programming task difficulty was found to result in a significant increase in students’ anxiety levels during the programming class [137]. Furthermore, the rising anxiety level when students face a programming error in their code could impact their ability to fix it [138]. Moreover, computer programming performance was found to be negatively related to mathematical and programming anxiety [139]. Unlike many people’s assumptions that CS students are very comfortable dealing with computers, Doyle et al. [140] stated that even final-year CS students might feel anxious when using a computer to complete some tasks.

Nolan and Bergin [136] reviewed three articles to find out if test anxiety affects programming students and concluded that it does have a negative impact on the students’ learning. The results showed that CS students, especially programming students, could experience a high to moderate level of anxiety associated with mathematics, tests, and programming itself, which could negatively impact their academic performance. However, no previous study has discussed the impact of foreign language anxiety on NNES students’ academic performance in programming courses that use English as the main medium of instruction. Therefore, this thesis focuses on examining this factor and its impact on first-year NNES students’ academic performance in

EMI CS1 courses. Since this factor has never been explored regarding the EMI CS1 course, it is important to review the studies that have been conducted in other courses and majors where EMI is applied.

Foreign language anxiety (FLA), as defined by MacIntyre and Gardner [141], is 'the feeling of tension and apprehension specifically associated with second language contexts, including speaking, listening, and learning.' Horwitz et al. [142] described a type of FLA that is mainly associated with learning and using a second language in classrooms, which can be measured. In their study, they referred to this type of anxiety as Foreign Language Classroom Anxiety (FLCA) and developed a scale to measure its level. By reviewing 97 studies conducted in 23 different countries, Teimouri et al. [46] found that FLCA is negatively correlated with students' language achievements in foreign language classes. Na [143] emphasises the impact of FLCA on students' learning achievements, stating that 'usually, high anxiety can make learners get discouraged, lose faith in their abilities, escape from participating in classroom activities, and even give up the effort to learn a language well. Therefore, learners with high anxiety often achieve low results, and low achievement makes them more anxious about learning.'

However, the impact of FLCA on students' academic performance has primarily been investigated in the context of foreign language classrooms. Very few studies address FLCA in a more general context, such as higher education programmes that use EMI in non-English-speaking countries. Suzuki [42] measured students' anxiety when speaking English at Japanese universities that use EMI, with the participation of 71 students. The result of this study shows that students tend to be more anxious about using English at the start of the semester, but their anxiety level reduces by its end. However, students who report a high level of FLCA stay highly anxious throughout the semester. One of the main limitations of their research is that they only considered measuring speaking anxiety, whereas FLCA involves several related components.

Their result is quite consistent with Cheng and Erben [144]'s study, which found that Chinese students' anxiety levels related to the English language decrease when they stay longer in the US, where they study. Their results were based on data collected from 156 Chinese graduate students, studying various majors, who were in their final four years of study at the time the research was conducted. These two studies Suzuki [42], Cheng and Erben [144] provide evidence of the existence of FLCA in a context wider than foreign language classrooms. However, they do not address the impact of this type of anxiety on the students' academic achievements in such courses. Moreover, in Cheng and Erben's study, all the participants were from the same cultural background and had studied in an English-speaking country for several years. There-

fore, the cultural and environmental impact might limit the applicability of the results in a broader context.

In a more specific setting, a study conducted on 78 students at the College of Nursing at a university in Saudi Arabia showed that nearly 70% were very anxious when communicating in the English language [43]. In the results, the fear of oral communication was attributed to two main issues: lack of fluency and pronunciation, while spelling was the main barrier for students when communicating in writing. In that study, the researchers investigated students' strategies to cope with using EMI but did not consider measuring FLCA level nor its influence on the students' academic performances.

A more inclusive study that included 187 students at a Business School at a South Korean university shows the negative influence of English language anxiety on students' achievements in EMI courses [44]. It also indicates that FLCA is related to class avoidance and more difficulties in understanding the course's learning materials than usual. Another research in an Indonesian university showed that non-English major students experience foreign language anxiety including apprehension in communication, fear of negative evaluation, test anxiety, and anxiety in English classrooms [45]. The result was based on data collected from 640 first-semester students in the Economics and Business schools.

Agarwal et al. [145] investigated the factors that contribute to differences in stress and sense of belonging between NES students ($n=606$) and NNES students ($n=348$) in three CS courses at a US university. The findings show that factors like 'personal life' and 'requirements for other classes' were reported more by NES students as a main source of stress. On the other hand, NNES students reported factors such as 'embarrassment asking for help' and 'self-doubt' as the main source of stress. In CS2, which is one of the three courses included in the study, NNES students found that the 'difficulty of the class' is a major factor that significantly impacts their stress levels. The researchers attributed this outcome to the high cognitive load that NNES students might experience while trying to comprehend difficult programming concepts in a foreign language. Even though some factors that affect stress levels in the NNES group seem to be related to language anxiety, the researchers did not address this type of anxiety directly.

Indeed, the majority of previous studies focus on the effect of FLCA in foreign language classrooms. However, the same source of anxiety exists in any other class that uses EMI. The reason behind this is that students in these classes use a second language to learn different subjects, communicate with their teachers and classmates, and complete the course assignments. As discussed in the previous section, one study showed that students might hesitate to engage in an EMI programming class if they were forced to speak in English, even though they reported having high speaking skills in Eng-

lish [146]. One possible explanation for this result might be the psychological barrier represented by FLCA, which students could experience in EMI courses, and which was found to be unrelated to English ability [147]. Nevertheless, there is a lack of research on the impact of FLCA on teaching a specific subject, such as programming, to students through a foreign language. Hence, this research aims to fill this gap in the literature by investigating the possible correlations between FLCA and NNES students' academic performance in CS1 classes. The correlation was expected to be found, not only because EMI was used in the selected CS1 classes, but also because the programming language itself is heavily related to English. Therefore, the review of the above literature on FLCA informed the following research hypothesis:

Hypothesis 3: Foreign language classroom anxiety level impacts NNES students' academic performance in CS1 courses.

2.4.7 Cognitive Load

The influence of cognitive load on learners has been a topic of research in education for many years. Van Gog and Paas [148] define cognitive load as 'the load imposed on an individual's working memory by a particular (learning) task.' Thus, the learner's academic performance in a specific task is directly influenced by the cognitive load needed to complete it [149].

As described by Schnotz and Kürschner [150], cognitive load theory (CLT) assumes that all learning processes occur in a part of the memory called working memory, which has a very limited capacity. This type of memory can only process four elements at the same time, and these can fade within 20 seconds. Conversely, permanent long-term memory has an unlimited capacity. Newly gained knowledge can be stored in long-term memory in the form of schemas, which can be used to solve similar tasks in the future. Each schema is relocated from long-term memory into working memory as one chunk when it is needed to facilitate a problem-solving process. Once a specific schema has been automated, the person becomes an expert at solving problems in that domain.

According to CLT, there are three types of cognitive load: intrinsic load, extraneous load, and germane load [151]. Intrinsic load is related to the difficulty of the new concept that needs to be learned, whereas extraneous load is the load caused by the design or presentation of this concept. Germane load refers to the mental resources required to create a schema of the new concept and store it in long-term memory.

Researchers have tried and adopted several methods of measuring cognitive load since the foundation of CLT. In fact, the difficulty of distinguishing between the three types

of cognitive load makes the measuring process complicated [150]. Also, the timing of the measuring process has a significant impact on the results [149]. For instance, measuring cognitive load can yield different results when done during and after the completion of the same task.

Morrison et al. [149] describes the adaptation of the Cognitive Load Component Survey, which was originally developed by Leppink et al. [152], for use in introductory computer science. This tool is one of the less expensive and complex measuring tools designed to measure the three types of cognitive load. The survey can be used to measure the cognitive load factors of specific lectures during the course. This tool is generally reliable for measuring intrinsic and extraneous load. However, it has been found to be less effective in measuring germane load, as well as in addressing issues related to self-assessment [153].

Bouvier et al. [39] state that NNES learners in programming classes could experience cognitive overload, as they have to learn a foreign language while also learning their programming subject. Students with low English proficiency may struggle with understanding abstract programming instructions like '*while*', '*for*', '*string*', etc., which may not have a clear meaning for them if they lack English proficiency. As a result, they will experience additional extraneous cognitive load while trying to understand the meaning of those terms, as well as their abstract meaning in programming [78]. The increased cognitive load could result in low academic performance in their programming courses or exams [10, 77].

By applying the CLT, Mason and Seton [41] analyse the exam results of international students studying information technology at Australian universities. Their findings support those of Bouvier et al. [39], as they discover that NNES learners often encounter a high cognitive load, which results in a high rate of failing final exams. Accordingly, the exam format was modified, and the students were provided with keyword glossaries in their native language, which resulted in a significant improvement in the students' performance. The research was repeated with a database course, where the exam was modified using the guidelines produced in the previous study, and NNES students were provided with a keyword glossary in several languages [21]. Compared to a technical course with the same group of students, their performance in the database course exceeded expectations.

The main aim of this thesis is to investigate the impact of the English language on NNES students' academic performance in CS1 courses. The focus of this research is on the students' academic performance in the complete course, not only a single lecture. Since each lecture can have a different level of cognitive load on the students, it is not possible to measure the total cognitive load of the course using the available

measurement tools. Moreover, this study was designed to provide more generalisable results by including students from two different universities in two different countries. Despite the importance of the cognitive load factor, it was not possible to include it in this study, as it requires a more controlled and specific research environment. Nevertheless, it should be stressed that cognitive load is an important factor that needs to be investigated in relation to NNES students' academic performance in CS1 courses. The study conducted by Kaur and Chahal [154] is a state-of-the-art and valuable resource for researchers interested in exploring this topic.

2.5 Native Vs. Non-Native English Speaking Students in Programming Courses

There is significant evidence that individuals who are not fluent in English may face considerable challenges learning programming when compared to native English speakers [124]. In fact, students who are not fluent in English were found to be more likely to fail or drop programming courses, especially when significant reading and writing in English were required [22]. This was one of the main findings of a study that examined the impact of students' characteristics on the outcomes of traditional and inverted offerings of programming courses. A total of 1,236 CS1 students at the University of Toronto participated, with the researcher using the midterm and final exam results, as well as the pre-course survey. The study's main limitation was the lack of a proper scientific method to measure and report the students' English proficiency, with participants only being described as non-fluent in the language.

Ivins and Ong [23] conducted a study that aimed to reduce the withdrawal and termination rate at an Australian university by finding the best predictor that will help in recruiting students into computing degree courses using data collected from 197 first-year CS students. They explored a number of different factors using psychological characteristics tests and background surveys. One of the main factors was the native language, and the results suggest that local students whose native language is English tended to perform better than NNES students in the computing courses' core subjects. However, the English proficiency level of the NNES students who took part in the study was not documented.

Kurtz [155] investigated the relationship between the development of abstract reasoning and performance in an introductory programming class. The study was conducted at San Francisco University, with 30 participants, seven of whom were NNES. The results showed that native speakers scored significantly higher on the intellectual devel-

opment test than NNES students, while there was no difference in programming scores between NES and NNES students. However, the total test scores for the two groups in the course were significantly different, again favouring native speakers. The researcher explained that the use of English in the test to measure intellectual development might have impacted NNES students' performance. The researcher also highlighted that this type of test could lead to an underestimation of NNES students' actual abilities when provided in English. Furthermore, the researcher noted that NNES students might have faced similar challenges in the written programming tests of the course, which were delivered in English.

Ford and Venema [156] evaluated the Dehnadi test, which is a short test aimed to examine if the students achieved the required understanding of basic concepts in programming after finishing the introductory course. The participants were 111 students in their second programming course who had already passed an introductory programming course at an Australian university. In the results, they noticed that students who were fluent in another language more than English scored significantly lower on the test than other students. This result is in line with what has been noticed by Mason and Seton [21]; even if it was not the main aim of their study, they found that international NNES students performed significantly lower than their local peers who speak native English in programming courses. The English proficiency level of the participants was not reported, which is one of the main limitations of the findings in both studies.

Guzman et al. [118] conducted a study that explores the differences between the experience of NES and NNES students attending four CS courses all considered introductory levels to programming in a US university. The results revealed that NNES students need to acquire higher marks in order to be satisfied, while they also show significantly lower average confidence levels than their NES peers. The results also show that NNES students spent significantly longer time studying on average outside of class but showed no significant difference in their scores in three of the included CS courses compared to NES students. Interestingly, NNES students scored significantly lower than their NES peers in one of the CS courses despite spending more time studying. These results are highly consistent with Sheard et al. [157] finding where NES students acquired significantly higher grades in a computer programming course than NNES students. This was one of the findings of a study that investigated factors influencing students' progression and performance in the first year of their undergraduate information and computer technology degrees. The study used records from 166 students who attended three courses: Computer Systems, Computer Programming, and Information Technology. English was the first language of 66% of the respondents,

based on a self-reported survey. Similar to most research in the field, the lack of reported English proficiency limits the generalisability of the findings.

NNES students may also face cultural differences that increase the challenges they face in EMI programming courses. For example, some cultures are strict when it comes to right or wrong, thus, students who have been raised in this type of culture may avoid asking questions or asking for clarification [40]. Therefore, they might misunderstand the assignments, which makes it more difficult for them to solve. The cultural barrier, plus the discomfort with English, were identified as the main reasons for NNEs students being unable to progress in programming courses on the same level as their peers who were born and raised in the US [40]. Carter et al. [40]’s study highlighted the importance of the cultural factor on students’ progression in a programming course and revealed interesting results; however, it suffered from a significant limitation: the very small number of participants, with only two NNEs students, and the failure to report their English proficiency level.

Johnson et al. [158] conducted an experiment to assess two code readability rules, namely nesting and looping, with learners from different levels of programming experience. The participants included 208 undergraduates in CS, 49 Master’s CS students, and 61 professional developers. The results revealed that native English speakers had significantly shorter reading and answering times. Additionally, the higher the participant’s knowledge of English, the more their readability and comprehension confidence ratings were affected by the minimise nesting rule. In a study conducted by Begel and Graham [159], differences in how NES and NNEs participants verbalised code were found to impact the interpretation of spoken Java, a variant of Java that can be verbally expressed. For instance, NNEs participants faced difficulties verbalising partial words (e.g. ‘tur’ and ‘pat’) or abbreviations, whereas native English speakers had no trouble with these. In another study, Guerrouj et al. [160] selected 50 random identifiers from an open-source C program and asked participants to split and expand them. The 63 students who participated in the study were from various levels, ranging from undergraduates to Ph.D. students and post-docs. The results showed that native English speakers and those with excellent English were more able to split and expand identifiers in code even with the existence of abbreviations and acronyms. Unlike most research in the field, which only categorises participants as native or non-native, the study assessed students’ English proficiency on a scale from ‘*bad*’ to ‘*excellent*’, although the rating was still based on self-evaluation.

Moreover, Li and Prasad [161] investigated students’ opinions on learning and accepting coding standards in three different programming courses using a questionnaire and observation methods. The total number of participants was 57, with 16 identified

as native English speakers and 41 as non-native English speakers. In the findings, the researchers stated that NNES learners often find it challenging to understand or adopt programming standards, which may be due to their difficulty in comprehending English-based standards. Regarding field-specific English terminology, it appears that NNES students are also at a disadvantage, even after years of study. Scott and Mason [94] investigated the challenges facing NNES students in the final-year Cybersecurity course by interviewing 31 students, 28 of whom were NNES. The results revealed that NNES students faced more difficulties with field-specific terminology compared to their NES peers. The main limitation of this study is that the researchers noted omitting seven interviews from the analysis but did not specify the number of NES and NNES participants included in the final analysis.

Prasad et al. [162] found no difference between NES and NNES students in representing programming concepts. The study included students who had completed two programming courses and were from different language backgrounds, 15 NES students and 14 NNES students. The researchers suggested that the organisation of programming concepts could be language-independent as the two groups represented concepts in similar ways. This result supports the findings of Feijóo-García et al. [163], who compared the performance of adult learners in block-based programming platforms, namely Scratch. The study included seven NES and nine NNES participants, all with no prior coding experience. The study concluded that the ability of the learners to come up with a solution for programming problems is not impacted by their native language, even though the NNES participants believed they would perform better if they used a programming environment that matched their native language. However, this result is in contrast with previous research that was conducted using data collected from the same programming platform, Scratch, from five countries and over 15,000 users [13]. The results indicate that novice programmers learn programming concepts faster when the tool interface they are using is adapted to their native language. Another research that showed contrasting results is a qualitative case study involving 29 NNES adults learning programming using different textual and visual programming languages and environments [164]. The researchers stated that the block-based environments were adapted to the learners' native language, and the programming keywords in the textual programming languages were translated and explained to them in their native language. However, the participants still reported that English was a barrier to their learning, especially in text-based programming languages.

In their doctoral thesis, Nnass [128] investigated the differences in the challenges facing NES and NNES students in programming courses. The participants were 103 NES students attending a programming course at an Australian university and 72 NNES

students attending a similar course at a Libyan university, although their levels of English proficiency were not reported. The results show that for NES students, the main issues were with loop statements (Do...While), all programming concepts, and program design. While for NNES students, error messages, pointers, and loop statements were identified as the main challenges. Other studies suggested that even though error messages are challenging for all novice programmers, NNES students will face more challenges dealing with those messages than those whose native language is English [110, 126, 123, 124], which supports Nnass's findings. However, in a study that examines the differences in compiler error distributions between NES and NNES novice programmers, the researchers found no significant differences [125]. The study used data collected from the BlueJ Blackbox database, an integrated development environment for the Java programming language, covering a full year. In the study, the English proficiency of the users was classified based on the English Skills World Ranking of the country they logged in from using their IP address. The researchers also reported another limitation: they categorised users as NES or NNES based on their IP addresses, which may not be accurate, as the country of residency does not always reflect the user's native language or their proficiency in English.

2.6 English Medium Instruction (EMI) in Programming Courses

English Medium Instruction (EMI) is the use of English to teach academic subjects other than English itself, which might have an impact on NNES students' academic performance [20]. Many universities around the world are adopting EMI, regardless of the native language of their students, in an effort to become internationally recognised institutions [47]. Computer science is one of the main fields where EMI is widely adopted. However, the number of studies on NNES learners in such classes is relatively small, despite efforts to broaden participation in this field [165].

Zumor and Qasem [166] found that EMI has a seriously negative impact on NNES students' academic performance in computer science, engineering, and medicine majors. The reasons were explained as using English in teaching those disciplines negatively impacting the students' comprehension of course content, as well as triggering negative feelings such as anxiety, frustration, tension, fear, and embarrassment. In the mentioned study, over 70% of participants believed that using English in course assessments was the main reason for their failure, while more than 85% felt they would perform better if assessments were in their native language. The study included 264

Saudi students from various majors at the same university. Data was collected through an online survey with open and closed-ended questions. However, the findings may have limited generalisability due to the absence of data on participants' English proficiency levels and their correlation with academic performance. Additionally, all students shared the same language background, and data was collected from a single institution, limiting the diversity of the research sample.

Raffaa and Ali [167] conducted a similar quantitative study with the participation of 200 CS undergraduate students studying at Saudi universities. The aim was to assess their attitudes toward EMI, the challenges it poses, and its impact on their learning experience. The results showed that while students recognise the global importance of English and have a positive attitude toward EMI, many struggle with English technical terms, which represent a barrier to comprehending CS concepts. The results also highlighted several challenges encountered by those students in EMI CS classrooms, including difficulties in writing, comprehending lectures, and communicating with teachers. Some students mentioned that they had lost their passion for computer science due to the difficulties they faced because of EMI. The limitations of this study are similar to those of the above study: the participants were from a single language background, and their level of English proficiency was not considered.

Another study investigated the experience of NNES students in the CS major at a university in the US [118]. The results revealed the struggles of this group of students, who had to study for significantly longer hours outside the classroom compared to their NES peers, yet their performance was inferior. Moreover, their confidence levels in EMI classes were significantly lower than those of their NES peers. Many studies support this result, concluding that when EMI is used, NNES students often score lower, unlike those whose native language is English [29, 168, 53, 11]. This is why policymakers have been encouraged to consider students' preferences for the language of instruction before implementing EMI, especially in non-English-speaking countries like Saudi Arabia [57].

Some studies have compared the use of the native language and English as a medium of instruction [168, 92, 169, 11, 170, 171, 172, 173, 174, 146, 175]. The debate over whether EMI or native language instruction is more effective remains a contentious issue. Pal and Iyer [170, 11] conducted a comparative study comparing the effectiveness of EMI versus native language instruction in computer programming courses using two different teaching methods: classroom instruction versus self-paced screencasts. Their research, which included pre-test and post-test performance assessments, demonstrated that students taught in their native language outperformed those in EMI settings. The researchers suggested that, for NNES students, a self-paced screencast

should be used. This mode of teaching helped the students perform better on the course compared to when class teaching was used under EMI. However, this study was conducted in a controlled environment and might lack real-world classroom dynamics. Moreover, the participants were from one linguistic background, which might impact the generalisability of the results. It is important to highlight that the two papers [170, 11], were work published as part of Pal [92]’s Ph.D. thesis and reported the same results.

Soosai Raj and co-authors [171, 172, 173, 174] conducted a number of research studies and experiments to explore the impact of using the native language of the students as a medium of instruction in different CS courses: a one-week programming course [171], Data Structures course [172, 174], and Operating Systems course [173]. They found no significant differences in the students’ performance between the group who attended bilingual classes versus those who attended only EMI classes [172, 174]. However, the students expressed a positive attitude toward the bilingual classes [171, 172] and tended to ask questions in their native language [171]. The bilingual teaching methods also showed a positive impact on the students’ engagement and interaction in the classes [173, 174]. The main limitation of all these studies is that all participants were attending EMI high schools, and the students’ proficiency in English was only reported in one of those studies [174].

A replication study was conducted to investigate the effects of Spanish-English bilingual instruction on high school students’ learning and engagement in a two-week CS0 course using block-based programming [175]. The experiment was conducted at multiple institutions in a rural area in the US. The study design included an experimental group that was taught using English and Spanish, which is the students’ native language, and the control group that was taught using only English. The researchers used a pre-test and post-test to evaluate the students’ programming knowledge before and after attending the bilingual instruction course. The students’ engagement in the course was measured by the number of questions they asked during the course. The results showed no difference in terms of students’ learning block-based programming considering the language of instruction, which supports the findings presented in [172, 174]. However, the experimental group showed more engagement with the course than the control group, as they asked more questions. In the experimental group, students were noticed to use English or English-Spanish in asking questions, not only Spanish. The researchers attributed these results to the possibility that students felt more comfortable and related to their instructors. The increased student engagement and the rise in class participation align with the findings presented in [171, 173, 174]. The main limitation of this study is that all students were from

English-language schools, therefore students in both groups were already proficient in English. The other limitation is that the course was only two weeks long and taught block-based programming. These two limitations might impact the generalisability of the results to students who are not fluent in English or other text-based programming courses.

Another study was conducted in Hong Kong with the participation of secondary school students: 81 students from Chinese medium of instruction schools and 138 from EMI schools [168]. The study aimed to examine the impact of the medium of instruction, as well as other factors including gender and prior academic ability, on student learning of computer programming. The researchers used an online survey and programming test to investigate the topic. The findings showed that students who learned computer programming through their native language outperformed their peers who attended EMI classes. It also showed that students with low and medium academic ability levels attending EMI schools were found to be at risk of failing the programming course. This result contradicts previous studies [172, 174, 175], which found no significant difference in students' academic performance when comparing instruction in the native language to EMI. The negative impact of EMI on NNES students was clear in this study; however, the results cannot be generalised to university-level courses due to the differences in participants' age and learning environment.

One more pilot study conducted by Khan and Nabi [146] could further complicate the debate about the medium of instruction. Their research compared the use of English and Urdu mediums of instruction for a university-level Object-Oriented Programming course in Pakistan with the participation of 33 Computer Engineering students. The researchers used a survey for self-reported English ability and a questionnaire with open and closed-ended questions exploring the students' preference for the language used in the course. In the results, the use of the students' native language (Urdu) was found to be preferred in oral communication, while English is preferred for written communication. Almost 40% of the participants stated that they would hesitate to ask questions if required to speak in English, despite reporting having a very similar level of English speaking skills compared to their native language. However, in the self-reported English proficiency level used in the study, students reported higher English writing and reading skills than their native language, which might have impacted the overall results. Although the students' perspective on the matter is highly important, this study could have benefited from investigating the impact of the language of teaching on the students' academic performance in the course.

Moreover, Grover et al. [176] study shows limited evidence that bilingual programming problem specifications could enhance the comprehension of NNES students. In

their study, Grover et al. presented a bilingual translation framework to translate programming problems from English to a combination of English and a regional language, which was Hindi in the experiment. In the first part of their study, a user study with 25 instructors was used to review the translation of a well-known programming problem. The second part involved 65 undergraduate NNES students studying programming who preferred Hindi or Hindi-English as their communication language and were randomly divided into experimental and control groups. The experimental group was provided with the translated English-Hindi (Hinglish version) programming problem, and the control group was provided with the English version of the same programming problem. By analysing the students' answers, the study concluded that there was no significant difference in the comprehension of the problem between the groups exposed to the English and Hinglish versions. However, the findings conflicted with the instructor survey; the majority of respondents supported the bilingual problem statements, indicating that bilingualism in instruction may have some advantages. Although the study was on an introductory programming course level, the researchers highlighted that all participants had studied at least one programming course previously, while some of them had also completed data structure or algorithms courses. Thus, the impact of the students' programming experience on their ability to comprehend the programming problem provided in the experiment remains unclear. The other limitations of this study include the small sample size and the focus on a single institution and culture, as well as the self-report of the English ability of the participants, which make it difficult to generalise its results. Moreover, the study relied mainly on a single programming task, which may not accurately measure students' comprehension in typical learning environments.

2.7 Factors that Correlate with Students' Achievement in Programming Courses

The understanding of factors influencing students' performance has drawn researchers' interest for many years. One of the earliest studies was conducted in 1917 to investigate the relationship between certain psychological factors and academic performance [177]. There are various research motivations behind this trend, which have changed over time. Early studies sought to identify the characteristics of individuals who would be good candidates for a particular task, such as programming [178]. Gradually, researchers shifted from identifying good students to understanding the factors that may influence students' academic performance [5, 6, 7]. Moreover, the de-

velopment of data mining and machine learning techniques has encouraged researchers to build prediction models for academic performance [8, 9]. This section provides an overview of the different factors that have been found to be correlated with programming achievements and, in some studies, have been used to predict students' academic performance in such courses.

Many of the factors that impact students' academic performance have been studied over the years. These factors can be categorised into four main types: demographic data, previous academic performance, psychological factors, and behavioural factors. The demographic factors include age, gender, race, economic status, and employment status, although many of these are less commonly used in recent research [179]. However, due to the trend of gender diversity in computer science research, gender remains an important factor [180]. Therefore, some studies on performance prediction have investigated its impact from different perspectives. For example, the impact of self-efficacy has been associated with low performance in programming; however, female students reported having a lower self-efficacy than male students as the course progressed [181].

On the other hand, previous academic performance in the same subject, or in different subjects, has been found to be a good indicator of students' performance in their current course. In CS, maths, science, and English have been identified as important factors in student success in different CS courses [70, 55, 182, 74, 31]. In this context, cumulative grade point average (CGPA) and in-class assessments are widely used in predicting students' academic performance [183]. Additionally, prior programming experience strongly correlates with students' academic performance in CS1 courses [10]. Therefore, Byrne and Lyons [70] suggested that programming experience should be considered in any research on academic performance in CS1.

As previously mentioned, psychological factors were the first set of factors studied, but they tend to appear more in psychology studies. However, the development of easy-to-adopt psychometric scales has enabled researchers in other fields to investigate these factors in their research [184]. For example, mathematical, computer, and test anxiety scales have been widely used to investigate the role of these types of anxiety on CS students' performance in programming in the computer education research field [136]. Moreover, the foreign language classroom anxiety scale (FLCAS) has been used to study the relationship between this type of anxiety and students' achievements in foreign language courses [185]. Other psychological factors that have been identified as playing a role in CS students' performance include self-efficacy, self-regulation, and confidence [179].

The use of learning management systems (LMS) has enabled researchers to collect huge amounts of data about students' behaviour while using these systems. For example, the pages visited, the number of times a particular video is played, participation in forums, and many other attributes have been explored by various researchers [146]. These behavioural factors have been successfully used to predict students' academic performance, especially in distance learning, where the education process depends mainly on students' interaction with the e-learning system [146]. Other studies have tried to predict CS1 students' academic performance based on their coding behaviour by analysing the IDE log data (e.g. [25, 186]).

Indeed, the factors that have been explored by researchers seem endless; however, most research has considered a small number of factors [179, 187]. Also, some research has investigated a set of factors that measure the same thing [179]. This has been mentioned in the literature as a disappointing fact about research in the area of predicting students' academic performance [179, 187]. Therefore, researchers have been encouraged to consider more factors in future studies. Similarly, there are few studies that use a combination of different data sources, such as psychometric scales, surveys, and demographic data [179, 188]. Moreover, a recently published literature review indicates that only two studies collected their data from more than one country, and very few others expanded their study to more than a single institution [9].

In this research, factors that influence NNES students' academic performance in CS1 courses have been investigated using different data sources, including a standard English language test, a computer terminology test, a foreign language classroom anxiety scale, and a background survey. Moreover, the data were collected based on two case studies, including two groups of NNES students attending CS1 courses at two institutions in two non-English-speaking countries over two semesters. While this study did not involve creating a prediction model for academic performance, the findings of the correlation analysis on the studied factors could greatly assist future research in this field.

2.8 Support Provided to NNES Students in EMI CS1 Courses

2.8.1 Curricular and Pedagogical Approaches

One of the most exciting studies in supporting NNES students in CS1 courses was by designing the curriculum using principles of Universal Design for Learning

(UDL) [189]. The Vantage One (V1) course was provided as an alternative entry option for students whose English language test scores were one band lower than the language requirements for the University of British Columbia (UBC). In this course, the students enrolled in the same CS1 course as other students who achieved the English language requirement, where the course was the same in terms of content but with a smaller class size. The students in V1 were also enrolled in a course called 'CS1-Content and Language Enrichment', which is specially designed to provide two hours of language support per week. This course involves a weekly one-hour CS1 language-focused class taught by an English language teacher and a weekly one-hour CS1 content-focused tutorial taught by a programming teaching assistant. To evaluate the new approach, the researchers compared the students' performance in programming assignments, two midterm exams, final exams, and the overall course scores. They also collected the students' feedback about the V1 course using a student feedback post-term survey.

The study was conducted over two academic years. In the first year (V1 $n=60$, Direct-entry $n=308$), students' performance in exams was not as good as their peers who did not receive language support courses. However, V1 students showed higher engagement in the course as they submitted more tasks. At the end of the course, the students in the V1 achieved a very similar overall performance to their peers who did not need language support. In the feedback results, 80% of the students stated that the CS1-Content Tutorial was helpful, while only 48% said the same about the CS1-Language Class. The researchers used the feedback survey to refine the V1 course for the following year's experiment.

In the second year, based on the students' feedback and following the UDL principles, the researchers redesigned the 'Content and Language Enrichment' course. The participants were 63 students in the V1 class and 347 students in the Direct-entry class. The result was very similar to the previous year; however, the difference in exam performance was not significant. This means that V1 students closed the gap in exam performance and continued doing better in terms of programming tasks.

Indeed, the results of this study were very positive and promising for a more universal course design that helps NNES students. However, the process involved in the course design and the production of its materials required a lot of preparation and collaboration between members of the computer science and academic English faculties. The demonstration of the course also required collaboration and organisation between the two schools as well. The experiment in that study also included a small-sized class for V1, which the researchers believed might have had an impact on the results. In reality, many universities might not be able to allocate all these resources to support NNES

students. Extra language classes and smaller class sizes are not as easy as they seem to be, as they require a lot of resources.

Another successful experience in designing a CS1 curriculum to support students from different language backgrounds is a course called '*CS Bridge*' [190]. It is an open-source curriculum for a two-week CS1 course that can be easily localised to different cultures, languages, and local teaching practices. In that study, a model for co-teaching CS1 courses using the proposed curriculum across different countries was also provided. Over six years, the designed curriculum and the teaching model have been adopted to teach over 1,000 high school students in four countries across four continents. These countries include Colombia, the Czech Republic, Turkey, and Guinea. The study also includes the collaboration of over 100 lecturers and undergraduate teaching assistants.

The main contribution of their work is the '*CS Bridge*' course material that was designed to teach the Java programming language to novice programmers using simple language that can be easily translated into other languages. The course material includes assignments, a clonable website, slides, and a lesson plan that can be used by any teacher, including student-teachers who have no experience. Despite some cultural barriers, the study represents a positive experience in designing a cross-cultural, co-teaching CS1 course. Providing high school students, especially in their first programming courses, with materials in their native language and content more related to their culture might encourage many of them to pursue their learning in CS at the university level. The authors stated that the main material of the course is in English and is provided by a program powered by Google Translate and adjustable by human translators, which can translate the entire course into different languages. The whole course has already been translated into three languages: Spanish, French, and Turkish. However, it is not clear if the translation tool has been tested on other languages, considering that a human factor is used in the translation process.

Mamman et al. [191] presented the development of an e-learning platform specifically designed to teach the Python programming language to Hausa-speaking students. The main aim of designing this platform was to reduce the language barrier faced by NNES students, particularly Hausa speakers, when learning Python through English-based platforms. The platform employs a user-centred design that incorporates plugins like LearnDash and TutorLMS to manage course creation, track progress, and issue certificates, and was built using WordPress. The course content, which covers beginner level, was translated into Hausa with the addition of culturally relevant examples such as farming and trading to make programming concepts more relatable and accessible for the targeted audience. Interactive elements like quizzes and coding challenges were included to enhance learning. The challenge facing NNES students in finding appro-

appropriate programming learning material in their language has been mentioned in several studies, therefore, such a tool can be a great method to support those students in their learning. However, in addition to the limitation of the languages that this tool supports, the study did not provide any empirical evidence of its expected benefits. More research is needed to be considered on the application of this tool in enhancing NNES students' programming learning experience.

In terms of pedagogical approaches, an interesting experiment on delivering a 'Cloud Computing Technologies' course using a flipped learning approach for NNES students in CS was presented by Doyle et al. [192]. In this experiment, the students were provided with pre-recorded lectures to be watched in advance of the lesson and online slides. For the laboratory work, students were provided with links to online tutorials and reference material, access to source code, and detailed laboratory sheets. The flipped learning approach adopted in the experiment means that students were supposed to study the provided online materials and answer related questionnaires and essays each day. The lecturer, who is a native English speaker, would address students' questions and provide feedback during the interactive laboratory work. In general, the students' satisfaction rate with the teaching method, learning materials, and assessment was very high. However, the small sample size ($n=10$) as well as the self-reported English proficiency level by the participants might limit the generalisability of the results. As stated by the authors, one of the main limitations of their study could be the motivation of the students to participate in the experiment. The majority of the students had joined the module because it was taught in English by a native English speaker in the hope of improving their English. In fact, the result of this study contradicts what was noticed by Horton and Craig [22], where students with poor English abilities were found to be more likely to fail or drop the inverted programming class because it requires more reading and writing in English.

Moreover, Pal [92] in their PhD thesis compared traditional classroom teaching with screencasting as well as comparing EMI with bilingual language instruction in teaching programming to NNES students. Their research aimed to build various scaffolds to teach programming to NNES students in different teaching methods and instruction languages. Using simple language and providing clear explanations for field-specific vocabulary were the main language-based scaffolds that were suggested to reduce the mental effort needed by NNES students in programming classes. Pal also recommended using self-paced screencasts for NNES students in EMI courses because this approach allows students to learn at their own pace, review content multiple times, and better understand the material without the pressure of keeping up with live class instruction that is provided in a foreign language. This teaching approach helps mitigate

language barriers that some students might face by providing them with the flexibility to pause, rewind, and focus on difficult concepts, thereby enhancing their comprehension and overall learning experience.

Xiong et al. [193] shared their experience in the implementation of a bilingual CS course at Chinese universities that uses English and Chinese. The project included developing a bilingual Data Structures and Algorithms course along with suitable teaching practice. The presented project is part of a larger reform project launched by the Chinese government that aims to improve higher education and includes selecting 500 courses to be provided in a bilingual mode. The authors claim that bilingual teaching can improve the students' proficiency in English while enabling them to comprehend the technical concepts more effectively in their native language. However, this paper lacks empirical data on the effectiveness of such a teaching approach; therefore, it remains in question how this kind of teaching influences students' competence in both languages and the subject studied. Meanwhile, the researchers highlighted many obstacles in adopting this teaching approach, including the availability of high-quality learning materials, the English language skills of the teaching staff, and the suitability of the teaching methods used in class. It is important to emphasise that, according to the authors, developing this bilingual module for one subject required about 10 years of development and still relies mainly on the availability of teachers who have high proficiency in English besides the field-specific qualification.

Another example of applying bilingual teaching, which is also from China, was presented by Li et al. [194]. The researchers proposed a level teaching model for teaching bilingual courses, including English infiltrator, English-Chinese combined, and fully English. In the first few lectures, the teacher used Chinese for most of the class but introduced the new terms in English. In that phase, the students were directed to read the English textbook, which has a Chinese translation edition. The second level was the use of a combination of English and Chinese equally in the class, but all the material provided was in English. The final level, which is called the Leading Level, was based on using only English in class and learning material. According to the authors, the results showed that using this approach of layering in introducing English in the course was helpful for the students to effectively learn the course material as well as improve their knowledge of English terminology. The authors stated that the designed course and teaching approach were tested three times with the participation of over 200 students each time, but no empirical data on student outcomes and feedback was presented or discussed. The lack of details about the evaluation process of this approach is a major limitation in assessing its effectiveness.

Jiang and Zhao [195] also presented an example of developing a teaching practice for a bilingual course, particularly designed for teaching Java programming in China. They provided a short description of what is needed for such courses, including teachers' training, material selection, used teaching software, lab exercises, and assessment methods. The main idea of this approach was to use English as the primary language for technical terms, documentation, and coding syntax, and Chinese was used to explain complex concepts and facilitate a deeper understanding of programming concepts. In the results, the researchers recommended using the ratio of 1:3 between English and Chinese to help students adapt to bilingual teaching. The research also emphasised the benefit of bilingual teaching in helping students master the subject and improve their English. Nevertheless, the authors failed to present their research methodology and provide the statistical data that supports their research results.

While the theoretical benefits of bilingual teaching are well articulated in the previous studies [193, 194, 195], there is a lack of empirical evidence that supports these claims, such as data on student performance, comprehension, and language acquisition. In the absence of this data, it is difficult to assess whether the approach significantly improves learning outcomes compared to EMI or another monolingual method. On the other hand, the challenges facing the adoption of bilingual teaching are quite difficult to overcome. This approach still needs a lot of development in terms of structure, materials, students' assessment methods, and managing the language used by students. It also requires programming teachers to be fluent in two languages, and it can only work if students are all from the same language background.

Hagiwara and Rodriguez [196], in their work, reviewed several related studies and suggested different factors that need to be considered when teaching CS to NNES students at the K-12 level. They stated that traditional CS instruction often fails to accommodate the linguistic needs of English learners, exacerbating disparities in learning outcomes. The teaching approach they describe emphasises culturally responsive pedagogy, integrating language support into CS curricula, and promoting inclusive practices that consider the diverse linguistic backgrounds of students. However, their research was mainly theoretical, and no real-world examples were provided to prove the usefulness of these strategies. The possible reason for that could be the fact that the practical implementation of the proposed strategies is challenging. Integrating language support into CS curricula requires significant resources, including teacher training, curriculum redesign, and ongoing support for NNES students, who could be from a mix of different cultural and language backgrounds. Additionally, the effectiveness of these strategies would depend heavily on the commitment of educational institutions

to prioritise equity and inclusion, which may not always be feasible in under-resourced schools.

In terms of educational theories, Liang and Ian [197] introduced the use of Multiple and Multidimensional Transitions (MMT) theory to design a comprehensive set of strategies that address NNES students' needs and facilitate their smoother transitions into EMI computing education. This was presented in an experience report on EMI-based Computing Education in a faculty of Engineering in Japan. MMT theory emphasises addressing the academic, socio-cultural, and psychological transitions that students experience as they adapt to new learning environments such as EMI. In the context of EMI and according to this theory, the academic transition involves the adjustment to new learning styles, understanding technical content, and overcoming language barriers. To ease the academic transition, the researchers incorporated strategies such as L1 support by allowing students to use their native language in class, exercise-based scaffolding, and gamification. For instance, L1 support helped students understand technical concepts without the added burden of struggling with English terminology. Scaffolding exercises were used to break down problem-solving into smaller, manageable steps, helping students incrementally build their understanding of programming concepts.

The second aspect of MMT is the socio-cultural transition, which refers to the students' need to adapt to new cultural norms, social interactions, and expectations, especially since some EMI environments are multicultural. To support students' socio-cultural transition, the researchers used peer interactions and extracurricular activities. A student mentoring system was promoted, where senior students helped juniors navigate both academic challenges and cultural differences. Additionally, multicultural events and clubs were organised to help social integration between domestic and international students.

The third dimension of MMT is the psychological transition, which encompasses changes in students' attitudes, confidence, and self-efficacy as they cope with the pressures of the EMI teaching environment. Considering that language barriers could cause emotional distress and reduce students' confidence, the researchers implemented strategies aimed at building self-efficacy and motivation. Activities such as fishbowl discussions and Kahoot! quizzes that encouraged active participation were used to increase students' motivation and help develop self-efficacy.

By mapping their interventions to these dimensions of the MMT theory, the researchers argue that addressing all three areas holistically will better support NNES students in their transition to EMI-based education, particularly in CS. Based on the details provided in the report, the theory seems to provide a comprehensive framework for

designing educational strategies that not only focus on language needs but also on broader student development and adaptation. One limitation of the study is the lack of empirical data to demonstrate the expected positive impact of this strategy on the academic performance of the students and their perspectives on its application.

2.8.2 Specially Designed Exams

Alhamami [57] in their study found that NNES students' GPA in EMI computing undergraduate programmes is highly influenced by their English proficiency. They suggested that educators need to find a way to assess students' knowledge of CS subjects separately from their English proficiency. The studies discussed in this section focused on database courses, not CS1 courses, yet the approach used is valuable and unique in assessing NNES students in light of their English language challenges.

In the first study, the researchers decided to analyse the exam script and the students' papers when they noticed a high failure rate, 48 out of 51, among Chinese students in an EMI database course at an Australian university [41]. The analysis results highlighted four main problems: not answering sufficient questions, avoiding highly interactive questions, and misspelling simple English words and keywords. The researchers attributed these results to the high cognitive load that NNES students face during exams, as many students showed an adequate understanding of the subject in the questions they attempted. The students' English proficiency levels were not reported nor considered in the analysis. Following this, the researchers redesigned the exam questions based on the Cognitive Load Theory (CLT) guidelines to reduce complexity and minimise cognitive load as much as possible. The students were also provided with a specially designed English-Chinese glossary to use during the resit exam. The students achieved significantly better results in the redesigned exam compared to the original exam. The results showed a great improvement in the pass rate from less than 6% to over 80%. However, nine students out of the 49 who attended the resit exam still did not manage to pass.

In a replication of their previous study, Mason and Seton [21] investigated the performance of NNES international students in a database course and a comparable programming course at an Australian university. The exam in the database course was designed using CLT principles, and the students were provided with a glossary in their native language, as well as the option to create and use their own glossary during the exam. The reviewed exams ensured the removal of any unnecessary information, using simple and consistent language in all questions, minimising ambiguity by changing any word that could be misinterpreted, splitting any question with multiple parts

into one-task questions, and using bold text to highlight the goal of the questions. On the other hand, the programming course was a regular course, and no enhanced exam or glossary was provided. The results favoured NNES students in the database course, where no significant difference was found between their performance and that of their local NES peers in the same course, as expected. The international students ($n=70$) had an average exam score of 70%, which is just above the average score of their local peers ($n=13$), whose average score was 69.4%. Unlike the database course, NNES students in the programming course, where no support was provided, performed worse ($n=89$, average exam score 46.4%) than local students ($n=16$, average exam score 62.7%). The difference in the performance of international students in the two courses was significant in favour of the database course's NNES students. One limitation of this study is the lack of reported English proficiency for international students, which could have impacted their performance in the included courses.

The adoption of CLT and glossary usage in the above studies showed positive and promising results in supporting NNES students. However, these supporting methods have only been examined in database courses provided in an English-speaking country, and no similar experiments have been recorded in CS1 courses. Thus, the results cannot be generalised to other courses or EMI environments in non-English-speaking countries. Moreover, it is difficult to judge the effectiveness of the adopted approach, considering the lack of information and consideration of the students' competency in English in those studies.

2.8.3 Non-English-based Programming Languages

Programming languages like COBOL, FORTRAN, and early versions of BASIC were designed with English keywords, reflecting the language's dominance in science and technology [15]. However, as programming spread globally and with the increase in the need for more skilled programmers, the limitations of English-based programming languages became more apparent. Researchers and educators in non-English-speaking countries began exploring alternatives to reduce the cognitive load and language barrier associated with learning programming by NNES learners. Programming languages in learners' native languages seemed like the logical solution to overcome the barrier of English and promote the comprehension of programming concepts for a broader audience. As a result, many programming languages based on different human languages have been developed, such as Arabic [198], Bengali [115], Portuguese [114], and many others. These languages were designed to align with the learners' native language, including the alphabet, the direction of writing, the punc-

tuation system, and the structure to reduce NNES learners' challenges when learning programming. Thus, most of these programming languages were mainly aimed at improving the teaching of programming to NNES students, not for advanced programming purposes.

Swidan and Hermans [199] describe a framework to localise programming languages, considering 12 aspects that can be localised. The proposed framework aims to help programming language designers in their localisation efforts for programming languages in different natural languages. These 12 aspects include alignment with existing language, keywords, variable names, productions, numerals and numbers, characters 'without meaning', diacritics, alternative keywords, localised punctuation, right-to-left writing direction support, multilingual programming, and error messages. The researchers applied the proposed framework to evaluate 12 localised programming languages, including block-based languages such as Scratch and Snap! as well as text-based languages such as Ebda3 and Hedy. The results indicated that although each of the evaluated programming languages had successfully localised some of the aspects presented in the framework, none of them managed to localise all of the identified aspects. Interestingly, numerals were found to be rarely supported, even though the localisation of this particular aspect is much easier than localising most other aspects, such as keywords. Therefore, efforts to provide NNES students or programmers with a fully localised programming language have not yet reached the expected level.

Despite the limitations of non-English-based programming languages, they have demonstrated some advantages over English-based programming languages, particularly in education. For instance, some studies concluded that programming in the learners' native language showed a faster and better understanding of programming concepts than programming in an English-based language [17]. This advantage has been attributed to the lower cognitive load that can be caused by using the learner's native language in programming compared to using a foreign language. Many NNES students reported reading and writing English-based programming code as one of the main challenges they face [10]; therefore, reading and writing code can be less challenging when the programming language matches the learner's native language. Another study on block-based programming platform users showed that NNES learners who programmed using an interface adapted to their native language developed programming skills faster than those who used an English interface [13]. Consequently, the use of the learners' native language in the IDE of non-English-based programming languages is expected to have the same benefits for students' learning processes. Moreover, learning a native-based programming language might help reduce the challenges reported in the literature related to learning English-based programming lan-

guages, such as spelling keywords, punctuation systems, and selecting clear identifier names.

When it comes to developing non-English programming languages, there are numerous challenges to overcome, with limited resources and support being one of the main issues. Most of these languages are the result of research efforts by interested scholars and are not embraced by major tech companies. The development and continuous improvement of such projects is a costly process that is difficult for individual researchers or small educational institutions to sponsor. It is understandable that large tech companies have little interest in these types of programming languages, as the potential profit from non-English-based programming languages may not justify the time and money invested in developing them, considering the relatively small audience they target. These programming languages are challenging to use in the industry, as many tech companies operate internationally and require standardised tech production and communication. Therefore, non-English-based programming languages could only be effective for learning purposes and cannot be developed or used on an industrial scale.

However, there are some drawbacks to incorporating non-English-based programming languages in education. Students might face challenges in transitioning from programming in their native language to more advanced English-based programming languages. Despite the similarity in programming concepts across programming languages, the language barrier becomes more prominent when students are introduced to English-based programming languages. In a study on transitioning students from using a block-based programming language to an English-based textual programming language, the results showed that one of the main challenges students faced was the English language [78]. There is a lack of research on the transition of programming learners from non-English-based programming languages to English-based programming languages, which might limit the understanding of the advantages and disadvantages of this approach.

Moreover, at the university level, students should be preparing to enter a global job market where English is the dominant language of technology and business. Universities might face the challenge of balancing the immediate benefits of non-English-based programming languages with the long-term need for students to master English-based languages. This presents an even greater challenge for universities with a wide variety of language backgrounds among their programming course students.

2.8.4 Translation of Code and Error Messages

Some researchers proposed translating or localising programming code to NNES students' native language to aid their learning. However, this is not an easy task, as there are too many programming languages and many more native languages. Piech and Abu-El-Haija [200] presented what they claimed to be the first tool to translate code between human languages. *CodeInternational*, as it was called, translates the comments, variables, and function names written in Java or Python code into the chosen human language. Using this tool, the programming language keywords and identifiers imported from libraries are not translated. The tool depends on 'Google Translate' in the translation process, which supports 100 spoken languages. This approach is supposed to help students from different language backgrounds in code reading and comprehension. The main limitation of this tool is not considering keywords in the translation, which is one of the main difficulties facing NNES students in learning programming [60]. The other limitation is the use of abbreviations for variable and function names in many codes, which might not be recognised by the translation tool used.

Perera and Ahangama [201] proposed a methodology of 'Language Switching', which basically gives learners the ability to view the code in both English and a chosen native language. Unlike *CodeInternational*, this approach includes the translation of programming keywords but does not translate the identifiers and function names. The researchers tested their proposed mechanism by applying it using Sinhala keywords, one of Sri Lanka's native languages, which showed its potential. However, the main limitation is the need for manual translation of programming keywords. The researchers highlighted that automated translation in this approach has a high risk of mistranslation and loss of context. Moreover, not translating identifiers and function names might limit the students' reading and comprehension of the translated code.

Instead of translating the code, Roehner [202] described a method of translating error messages generated by any compiler to any natural language. The method depends mainly on 'Google Translate' and works as a word-by-word translation. Although the authors outlined a framework for translating error messages, they did not sufficiently address the practical challenges of implementing such an approach. For instance, ensuring the consistency and accuracy of translations across various programming languages and environments can be a significant challenge, particularly due to the variations in the interpretations of technical terminology in different contexts. Another limitation of their work is that some abbreviations and technical words that could appear in the error messages might not be recognised or properly translated by the pro-

posed translation tool. Moreover, the complexity of error messages is usually the main issue, not only the language used [123], which has not been addressed in their work.

Carvalho et al. [131] in their research worked on enhancing the readability and translating Python compiler messages into Portuguese. They reduced the amount of jargon and complex vocabulary and simplified the structure of the message before providing the translation. The authors compared the traditional Python error messages with the enhanced versions, finding that students were able to understand and resolve errors more effectively with the improved messages. However, the applicability of the enhanced error messages to other programming languages was not discussed, which might limit the generalisability of the study. Nevertheless, the research on enhancing and translating error messages is promising but still needs considerable effort to be usable and accessible by all learners, considering the large number of programming languages and human languages that need to be included.

Dunn and Ghosh [203] introduced a translation tool for Python error messages called BicePy, a VSCode extension designed to provide bilingual error explanations, primarily aimed at helping NNES programmers, especially Spanish speakers. BicePy uses Neural Machine Translation (NMT) to translate error messages and offers context-aware descriptions via a Generative Large Language Model (GLLM). The authors claim that this tool aims to improve accessibility and inclusivity in programming by simplifying the understanding of complex error messages for NNES users. However, BicePy has some limitations that could impact its effectiveness. While it can operate in offline mode using a local database of common Python errors, full functionality, such as detailed explanations and context-aware descriptions, still requires an internet connection. Additionally, though the researchers suggest that the tool could be scaled to support other languages, no actual implementation for languages beyond Spanish was demonstrated. Despite these limitations, BicePy holds great potential to enhance comprehension, reduce cognitive load, and improve the overall learning experience for NNES students in programming classes. However, the evaluation of the tool focused primarily on translation quality through metrics like BLEU and chrF scores, without assessing user experience. This omission is significant, as user experience is essential to fully demonstrate the tool's usefulness in real-world educational contexts.

2.8.5 Computer Terminology Teaching

Mastering field-specific terminology is one of the most important language components that CS students need in their studies [99]. When comparing EMI with native language teaching, students expressed their desire to continue with EMI, but they

required verbal explanations for terminology in their native language [92]. In fact, the availability of explanations of English terms in the student's native language can help with better understanding and faster comprehension of the learning material, especially for those with low English proficiency [204]. Diethelm and Goschler [86] published one of the early studies that tried to raise awareness of the importance of clarifying and discussing new terms in CS class. According to the researchers, this can be particularly important for NNES students, and the suggested method is using 'meta-discourse' for new terms. In this method, students are encouraged to discuss the syntactic and semantic features of the terminology and its formal and informal use. Educators are also recommended to clarify terminology and ensure that students understand the precise meanings of terms used in computer science class. These recommendations are grounded in the realities of classroom teaching and could be directly applied to improve educational outcomes. However, Diethelm and Goschler [86]'s work is mainly theoretical and lacks empirical application that supports the proposed approach.

Different teaching methods and technologies have been adopted in CS education to facilitate the learning of new CS terminology. For instance, Chan [76] investigated the impact of pre-teaching technical vocabulary to NNES students in introductory information technology classes. In this experiment, the participating students were divided into two groups: the treatment group ($n=14$) and the control group ($n=33$). The students in the experiment group were offered two terminology classes where specific Microsoft Visual Basic vocabularies were presented and explained with examples and hands-on exercises, while the control group used a self-learning approach with the help of a prescribed manual. The results showed a significant difference in the learning outcomes in favour of the treatment group. This study provided empirical evidence of the benefits of pre-teaching technical vocabulary that can enhance learning outcomes. It also offered a detailed explanation of the terminology classes and the process of the activities included. However, the small sample size and the short duration of the experiment, covering only four weeks out of the 12-week course, could have impacted the results.

In terms of tools, Martínez-Valencia et al. [205] proposed an app to help CS students and professionals learn English CS-specific terms by providing translations in the learners' native language. The terms available in this app were 1,606 words from CSAVL, a specific CS word list previously published by Bi [206]. The app targets Mexican users (Spanish speakers) only and uses Google Cloud Translation services, with the help of seven CS experts to select the best translation for the app. The app design was mainly based on a flashcard methodology, where each card showed the English

word, its translation, and an example sentence. To evaluate the app, the researchers used English vocabulary tests (pre-test and post-test) with the participation of 32 individuals from fields related to CS, with varying levels of English proficiency. The results showed that the app helped users learn around 36 new words on average per week, using it for only 10 to 20 minutes per day. The learning rate was also unaffected by the users' proficiency in English, providing equal learning opportunities for all users regardless of their English level. Despite the study's limitations due to the small sample size, it provides a practical example of how to help NNEST students learn new technical words in a short time. However, developing such apps in different natural languages requires significant effort and cooperation. Moreover, this approach does not include activities like peer review, feedback, or discussion, which might limit its learning benefits.

Another study presented a game-based learning method to teach CS terminology to primary school students [207]. It involves designing and evaluating an interactive computer game called '*Rusty Rusty*' to teach IT terms in three languages: English, Russian, and Kazakh. In the design of the game, the researchers used a list of what they claimed to be the most commonly used terms in the field of IT. In the evaluation process for the developed game, the researchers collected qualitative and quantitative data from 100 students. The participants were divided into two groups: a control group and an experimental group, with 50 students in each group. During the observation, most students in the experimental group demonstrated active listening, focused attention, made eye contact, and actively engaged in the course, as well as responding to teachers' suggestions. They also showed more use of decision-making and problem-solving skills. In the feedback survey, the majority of students were satisfied with the benefits of applying this game to their learning, and they stated that the game enhanced their enjoyment of the class. The administrative data collected in the study also showed that the experimental group had a higher attendance rate, with fewer students being late for class compared to the control group. This approach to teaching computer science terminology showed promising results in enhancing schoolchildren's learning process. However, the game was designed for primary school students, which limits its applicability to a certain age group, and the results cannot be generalised to university-level students. Another limitation is the languages supported by this game, as well as the ambiguity regarding the process of selecting and translating the terms used in the activity.

Another experiment conducted by Da Silva and Silveira [208] investigated teaching technical terminology using a digital game with undergraduate students in the Fundamentals of Programming course. The game was a modified version of an open-source,

free-licensed digital entertainment game, with the main change being the addition of educational content alongside other adjustments. 42 undergraduate students from a computer science course participated in the qualitative evaluation of the modified game. The results showed that about 85% of the students agreed on its usefulness. The game also appeared to have a positive impact on the students' motivation and interest in learning programming. The main limitation of this study is the lack of information about the educational content that was added to the game and how it was prepared. Moreover, while the paper discussed the audience's background as being Brazilian, it is unclear if the designed game included any translated material into the students' native language.

Another example is the work conducted by Simanjutak [209], where they evaluated the use of Mobile-assisted Language Learning (MALL) to teach specific academic English vocabulary to CS students. The selected app, called SPEARA (Specific English as Research Assistant), is a mobile application aimed at helping students learn and use English in their academic majors. The app is funded by the Indonesian Higher Education Directorate and is available free for all students. The researchers conducted a pre and post-test using a list of the 50 most frequently used words in CS, selected from textbooks. Both tests contained two sections: one where students identified their familiarity with the words, and another where they wrote a sentence using each word. In the experiment, students completed the pre-test, then engaged in nine weeks of independent learning using the SPEARA app for the words included in the test before participating in the post-test. The results showed improvement in the students' knowledge of CS terminology; however, the interviews revealed that students believed MALL could not substitute human interactions. The researchers concluded that although the use of mobile applications in teaching specific field terminology is promising, the importance of collaboration and peer review should be considered in future studies. The main limitation of the study is that all participating students had a minimum of intermediate-level English proficiency, which may have impacted the results. Another limitation is the lack of information about the process and resources used in selecting the words for the experiment. Moreover, as the mobile application evaluated in this study was not developed by the researchers, there is a lack of details on the development process, which could have been useful for future research.

Wahyuni et al. [210] proposed an interactive web-based module for English for Specific Purposes (ESP), with a particular focus on English for Computer Science (ECS). The authors argue that existing ECS materials lack relevant content, making the subject less engaging for students. The proposed module includes multimedia elements such as audio, video, and quizzes, allowing students to learn at their own pace, with the ad-

ded benefit of being paperless to support Sustainable Development Goals (SDGs). This module is suggested to be particularly useful for NNES students, offering a dynamic learning environment that supports their acquisition of technical terms. However, the paper does not present practical steps on how this learning tool could be built or evaluated for effectiveness. While the theoretical framework is strong, further research is required to investigate the practical implementation and determine its impact on students' learning outcomes, particularly for NNES learners in computer science.

Indeed, the involvement of students in the learning process is one of the most effective methods of learning [211]. Studies have confirmed that students can be more creative when given the chance to collaborate with their peers, which can also enhance productive peer and self-assessment [211]. In programming courses, it is useful to adopt collaborative learning methods, which can benefit students as individuals as well as their collective learning and growth as a group in the course [212]. This thesis investigates whether an online managed and peer-reviewed translation activity for CS1 terminology can positively improve student knowledge. The activity also includes collaborative work to build a glossary for the course by the students. At the end of the semester, the students' opinions about the advantages and disadvantages of this activity, as well as their suggestions to improve it, were also considered. More details are presented in chapter 6.

2.9 Literature Context, Gaps and Weaknesses

Previous studies that focus on NNES students in university-level CS1 courses are limited, which explains the consideration of a wider scope of literature in this chapter. This section aims to review the research efforts that have been conducted in this area over the past 23 years and highlight the gaps and limitations in the knowledge. A total of 94 research publications published from 2001 until the end of 2023 were identified; Figure 2.1 shows the number of publications by year. This includes all publications that have NNES participants from all ages and school levels and focus on either programming or other closely related subjects in formal and informal learning. Some of these studies described programming tools and new programming languages that have been evaluated by NNES participants. The list and analysis of the included publications can be found in Appendix D.

Although researchers tried to raise awareness of this topic more than 20 years ago, it is clear that it has only attracted attention in the past three years. The number of studies that included NNES participants has certainly increased, but only 31 studies have

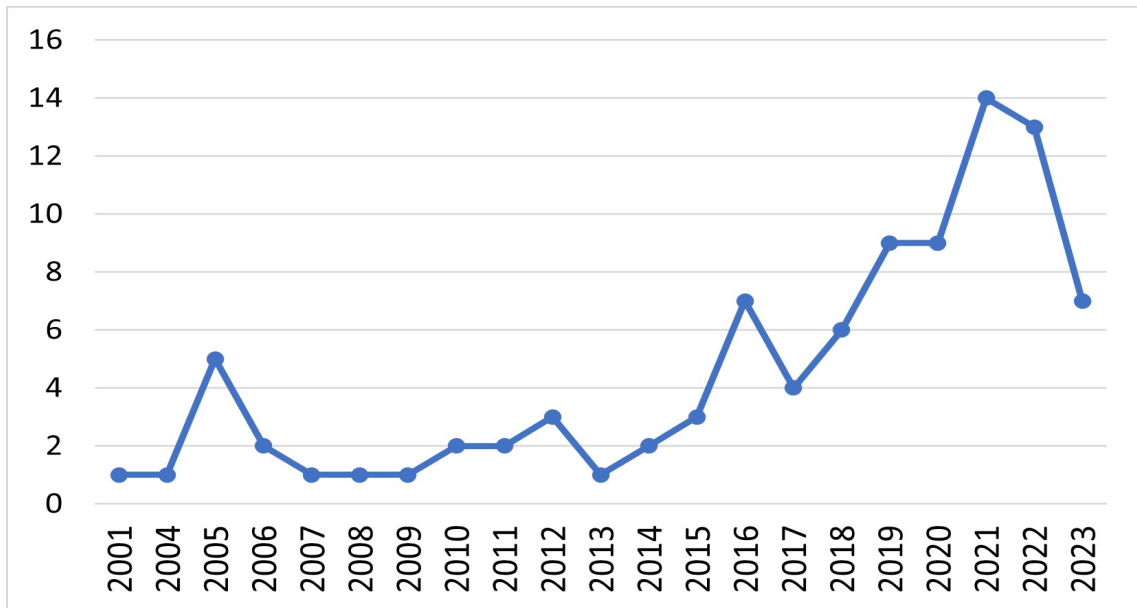


Figure 2.1: The number of research published from 2001 until the end of 2023 that included NNES Programming or CS Students in the participants.

been conducted on formal programming courses at the university level since 2001. Moreover, out of the 31 studies, only three of them discussed the difficulties that NNES students face in university-level programming courses [30, 128, 129]. In fact, two of those publications reported the same study; one of them was a PhD thesis [128], and its results were later published in a conference paper [129].

The first study by Mhashi and Alakeel [30] investigated the difficulties that NNES students face at a Saudi university, considering programming concepts, the learning environment, and learning materials. However, the impact of English on the studied factors was not directly investigated. For instance, to examine the effect of teaching materials on learning programming, the researchers asked students to rate the provided learning materials on a scale from 1 (practically useless) to 5 (very useful) but did not mention the impact of English. The researchers concluded that the surveyed students' level of English impacted their performance in programming based solely on their confusion between two English words that appeared in the programming test.

On the other hand, the thesis by Nnass [128] focused on identifying the most challenging programming concepts for NES and NNES students, including loop statements, pointers, arrays, etc. The researcher considered the challenge of English and highlighted the differences in the difficulties faced by both groups. However, the researcher did not measure or report the students' proficiency level in English, relying mainly on their classification as NES or NNES students. The sample size in that study is relatively small, especially for the NNES students (NES $n= 103$, NNES $n= 72$). Thus, this

this thesis aims to fill this gap in knowledge by investigating the difficulties that NNES students face in CS1 courses, focusing more on English language challenges and using a larger sample size. This thesis uses a standard English test to measure the students' proficiency level and other reliable tests to investigate other English-related factors impacting NNES students' academic performance in CS1 courses.

Four other studies, out of the 31, focused mainly on investigating the correlation between students' proficiency in English and their ability to learn programming [54, 72, 31, 56]. However, only one of them used standard English language test results to measure the students' proficiency in English, although the sample size is relatively small ($n=99$) [54]. Moreover, two different types of programming courses provided in different majors were included in that study. In this thesis, a larger sample size from very similar CS1 courses was used. In the other studies, English proficiency was measured either by students' academic performance in English at high school or through self-reported evaluations. Academic performance in English as a subject in school can be highly influenced by teachers' judgments and course structure, which makes it less accurate than standard tests. Similarly, self-reported evaluation of English proficiency can be highly influenced by other factors such as self-esteem and confidence levels. As mentioned above, this thesis uses a standard English test to overcome the limitations that appeared in many previous studies in this area.

The majority of the other studies (12 out of the 31) tried to identify factors that can predict or influence students' success in programming or CS in general, including the native language of the students or their English proficiency as possible predictors [70, 71, 29, 40, 157, 32, 156, 182, 65, 22, 73, 74]. It is important to mention that studies predicting students' performance in programming courses that did not consider English or students' native language are not included in this review. Again, none of these studies used a standard English test when investigating English proficiency as a possible predictor. Besides using previous academic performance records in English or self-evaluation, some studies only relied on students being native or non-native English speakers as a predictor. Although this was not the main aim of this thesis, it investigated the possibility of using English proficiency as a predictor for students' academic performance, as measured by a standard English test. It also explored whether the other studied English-related factors, including students' knowledge of computer terminology and FLCA, could be used to predict students' academic performance, which none of the previous studies had considered as predictors for students' academic performance in programming courses.

Only one study, out of the 12, explored the possibility of using the density of 'buzzwords' in learning materials as a predictor for students' academic perform-

ance [32]. The researchers explained that buzzwords are those with computer-science meanings but failed to show the process of selecting or identifying those words. This thesis is the first to revisit the correlation between computer terminology and the academic performance of students in CS1 courses, as mentioned by Keen and Etzkorn [32]. However, this thesis considers the students' knowledge of such terminology, rather than terminology density in the learning materials. Unlike Keen and Etzkorn's study, this thesis details the development and design process of a CS1 word list and a computer terminology test that was used to investigate this correlation.

Other nine studies investigated different types of teaching methods, including the medium of instruction, curriculum, non-English-based programming languages, and teaching tools in programming courses where NNES students were enrolled [11, 39, 92, 189, 114, 208, 127, 131, 176]. However, very limited work has been dedicated to helping NNES students improve their knowledge of the English computer terminology they need in CS1 courses. As a result, the practical part of this thesis focuses on introducing and evaluating an online learning activity designed to increase NNES students' knowledge of the computer terminology used in CS1 courses.

Two other studies compared NES and NNES students, considering the internal representation of computer programming concepts [162] and their experience learning computer programming at a US university [118]. Another study explored code readability rules, including nesting and looping, in two programming languages and revealed interesting differences between NES and NNES students' code-reading abilities [158]. The main limitation in most of these studies is that the number of NNES students did not match that of NES students. Additionally, these studies were conducted in English-speaking countries, and the students' proficiency in English was not reported. While the comparative studies on this topic provide valuable insights, this thesis focuses on NNES students, who are underrepresented in most of the previous research in this area.

Finally, one study explored the terms NNES students use to describe code snippets in their own language [101]. A key limitation of that study is that the participants were asked to explain the code in German, despite it being mentioned that some students came from different language backgrounds. It is unclear how the students' native languages and their proficiency in both German and English may have impacted the results.

2.10 Conclusion

This chapter highlighted the potential impact of English language proficiency and related factors on NNES students' academic performance in programming courses. Several studies have suggested that students with lower English proficiency often face challenges in these courses and show lower engagement in class activities. Therefore, it can be concluded that English proficiency is a significant factor that can directly or indirectly affect NNES students' academic performance in CS1 courses. However, no previous study has investigated this factor, along with related aspects such as English computer terminology and FLCA, using a large sample of NNES students at the university level, particularly in non-English-speaking countries that implement EMI.

The review and analysis of previous research on this topic identified four key gaps. First, no study has used a standard test to measure NNES students' English proficiency when investigating the challenges they face in programming courses. Second, the impact of students' knowledge of computer terminology on their academic performance in CS1 is unclear, and there is no standard method to assess this factor. Third, no investigation has explored the impact of FLCA on NNES students' academic performance in EMI CS1 courses. Fourth, no previous work has considered NNES students' perspectives on studying CS1 through EMI or studying English-based programming languages. Furthermore, despite the large number of NNES students attending EMI CS1 courses worldwide, very little has been done to address their specific English language needs. This thesis aims to fill these gaps in knowledge while addressing the limitations identified in previous research.

RESEARCH METHODOLOGY, METHODS, AND DESIGN

3.1 Introduction

This study applies a multiple case study research methodology and a combination of research methods to collect quantitative and qualitative data. This chapter starts by providing an overview of the theoretical framework of the study followed by the research chosen methodology. The research methods and design represent the heart of this chapter where they are presented in great detail. The data collection process and timeline as well as data analysis procedures are also described in detail. More information about the participants and the sample size is also available in this chapter. Finally, important ethical considerations related to this research were discussed and clarified.

3.2 Theoretical Framework

The theoretical framework sometimes referred to as the paradigm, is a frame of reference that guides the researchers to build and support their study [213, 214]. As defined by Guba [215], a paradigm is a basic set of beliefs that guides action. The chosen paradigm is used to explain phenomena, find connections, and draw conclusions. It also helps to identify and clarify perspectives and assumptions [216]. Consequently, the theoretical framework influences every decision and step the researchers make in their research process [214]. There are a number of theoretical paradigms identified in the literature including positivist and post-positivist, constructivist, interpretivist, transformative, emancipatory, critical, pragmatism, and deconstructivist. Those paradigms can be applied in different contexts and fields and from different

perspectives [216]. The focus here will be on post-positivism as a framework for this research.

The positivism approach is regarded as realism, common sense, and objective approach [217, 218]. It is sometimes referred to as the 'scientific method' as it aims to understand and explain the phenomena scientifically [219]. As described by O'Leary [220], positivism involves testing theory or explaining an experience using observation and measurement in order to 'predict and control forces that surround us'. As the main aspect of positivism, which is pure objectivity, is impossible to achieve, post-positivism accepts imperfections and tries to acknowledge them. In other words, positivism involves putting all personal beliefs and biases aside and studying the phenomena in a completely neutral way. On the other hand, post-positivism accepts the fact that no one can be completely unbiased instead it works on recognising and limiting the impact of such biases. Also, post-positivists consider their observations and findings cannot always represent the truth as they can also be subjected to error. Thus, the post-positivism approach encourages the use of multiple research methods to avoid unpredicted errors. Positivist and post-positivist research is most commonly used with quantitative methods of data collection and analysis [216].

The theoretical framework can be different based on the field of the research, which in this research is computing education research. Computing education research (CER) can be considered a multidisciplinary research field, mainly computing and education and sometimes social sciences and psychology [221]. CER can be defined as investigating the process of teaching and learning computing in order to design or improve teaching techniques [222]. The main goal of CER is to identify problems facing students in learning different computing topics and try to find new teaching approaches to overcome these problems [221]. In this research, the main goal is to identify problems facing NNES students in EMI CS1 courses, particularly problems related to the use of English, and find a way to improve their learning experience.

In the field of education research, the most commonly used approach is the constructivist approach [223]. In the same way, researchers in CER followed the same approach based on the similarities between the two fields [221]. However, Phillips and Burbules [224] and Panhwar et al. [223] suggested that post-positivism is the paradigm that is appropriate to be used in educational research as in other scientific research. They justified their view by explaining the advantages of post-positivism as a more scientific and flexible approach that allows the researcher to use multiple methods. The use of different instruments allows to minimise bias and errors and maximises reliability.

Post-positivism usually seeks to find correlations between variables while simultaneously considering participants' perspectives [218]. This research investigates the cor-

relations between three English-related factors and NNES students' academic performance in CS1 courses as well as exploring the students' perspective on the topic. Thus, post-positivism seems to be the most appropriate approach to undertake this research. Moreover, the learning needs of NNES students in CS are a topic that is still not well understood [165]. Therefore, it could be beneficial to adopt an approach that not only answers questions but can define the problem and find the right questions [225].

3.3 Research Methodology

Research methodology works as a roadmap for selecting research methods and analysis techniques to address the research questions and objectives. The case study methodology is one of the popular methodologies used by researchers in different fields [226]. According to Leedy and Ormrod [227], a case study is the most appropriate approach to investigate topics or situations that are poorly understood or explored. A more general definition for a case study is where the researcher explores in depth a program, an event, an activity, a process, or one or more individuals [228].

The case study approach is usually designed to use multiple sources of data to investigate a topic which enables forming new questions as new data arises [226]. Using multiple sources for extensive data collection leads to a deep understanding of this particular case under investigation [229]. Therefore, a case study provides researchers with the flexibility they need in selecting methodology and techniques to explore phenomena that have not been well studied in the past [230]. Thus, given the fact that there is a lack of research on this topic, the case study methodology was deemed the most appropriate approach.

There are two types of case study research: single case study and multiple case study. The main difference between those two approaches is that multiple case study allows for replications for the investigation of the same phenomena across two or more cases [231, 232]. In a multiple case study approach, the findings from the individual cases are expected to either be very similar in a literal replication or would have logical, explainable differences in a theoretical replication [232].

The use of a multiple case study approach enables comparison and replication of findings across different contexts which makes the findings of the research more robust than the findings of a single case approach and helps overcome the generalisability limitation [232, 233]. In other words, having more than one case not only extends the sample size but also allows cross-case validation whereby an outcome in one case can be replicated or differ from another. This cross-validation strengthens the generalis-

ability of findings because it demonstrates that results cannot be particular to a single context but rather are replicated across different settings [233]. While increasing the number of cases improves generalisability and transferability, it should be noted that the aim of transferability cannot be absolutely ensured but can only be applied carefully with respect to case-specific characteristics.

In this research, the first case study includes Chinese students in EMI CS1 courses, providing insights that may be informative for other NNES students, especially those from East Asian countries whose languages share certain linguistic features with Chinese. Languages such as Japanese and Korean partly share grammatical structures with Chinese and also use logographic or syllabic scripts [234]. This is in complete contrast to the Roman alphabet, on which English-based programming languages are built. Even though the writing systems of Korean and Japanese are dissimilar from Chinese characters, these languages also require that students adapt to a completely foreign alphabet when learning English-based programming languages. Moreover, each of these three languages has its own unique punctuation and sentence structure, which could affect how a student will view syntax in coding. The left-to-right horizontal writing direction in English differs from the traditional top-to-bottom vertical scripts used in certain East Asian contexts, which may introduce additional challenges. Thus, the results of this case study in Chinese might have an implication for other East-Asian learners, like Japanese and Korean students, who may face similar linguistic and cognitive difficulties while engaging in English-based programming language courses that use EMI.

The second case study included in this research was performed in Saudi Arabia with a population of students who were native Arabic speakers. The Arabic language is very different from the English language in terms of alphabet, grammar, punctuation, and even writing direction. Arabic is read and written from right to left in a cursive format different from the left-to-right, non-cursive format of English and most programming languages. This may cause a cognitive overload with Arabic-speaking students who have to get used to this new direction. Moreover, the Arabic grammar and punctuation systems are unique, especially with respect to the symbols and the use of commas and periods. However, the results from the Saudi context might be generalised to students whose language is closely related to Arabic, such as Persian (Farsi) and Urdu. Although Persian and Urdu are unique languages, they are both similar to Arabic in that they all are written from right to left with a similar alphabetic system, carry some of the same grammatical structures, and use many of the same conventions in punctuation [235]. These shared characteristics include vocabulary and phonetic structures that may influence how they process English keywords and programming

syntax. With such linguistic parallels, a hypothesis may be reasonable that such students may also experience similar difficulties with EMI CS1 courses and the findings of this research could have implications for their programming learning experience.

In both case studies, the distinct differences between the native languages of the students and the English language — whether in terms of alphabet, grammar, punctuation, or writing direction — present significant challenges that are not only specific to the study's participants but also relatable to other NNES populations. Including two diverse linguistic groups in this research already strengthen the generalisability of the findings; however, the transferability goes even further and covers students whose native language possesses features of either Chinese or Arabic. Therefore, the finding of this research is relevant not only for the specific contexts of China and Saudi Arabia but also in areas where similar linguistic barriers might affect student engagement with the EMI programming courses.

In case study methodology, the case boundary should be clearly defined to keep the study within its scope and help prevent any ambiguity [226]. Defining case boundaries may include social group, geographical area, or time frame [226, 232]. In this research, the cases are bounded by first-year NNES students attending EMI CS1 courses at the university level. The selected cases include courses provided at two institutions in two non-English speaking countries, namely, China and Saudi Arabia. In this research, 'Case Study One' will be used to refer to students attending CS1 courses in China, and 'Case Study Two' refers to students attending CS1 courses in Saudi Arabia. The data for the multiple case study were collected over two academic years (2020/21 - 2021/22).

Although the structure, materials, and assessments used in both case studies were comparable, providing a strong likelihood of similarities in the findings, each case study maintained its own uniqueness. Notable differences between the cases were reported and discussed in the discussion chapter. These differences, alongside the similarities, are intended to offer a richer and deeper understanding of the studied phenomena. Consequently, the use of multiple cases enhances both the depth of insight into the individual cases and the potential generalisability and transferability of the findings, offering a broader view of the challenges and outcomes faced by NNES students in EMI CS1 courses.

3.4 Research Methods and Design

This research is designed to include three main phases: **Phase One:** research instruments development and data collection; **Phase Two:** data analysis for each case study

individually; and **Phase Three:** cross-case analysis and hypotheses testing. In order to gain a deeper understanding of the studied topic, this research adopted a mixed methods approach by collecting quantitative and qualitative data. Based on the findings of the multiple case study, an application to help the students participate in translation activities in CS1 class was developed and tested. Following is a detailed description of the three phases of the research, along with the research instruments and techniques applied in the multiple case study. The process of developing and testing the translation activity application is presented separately in chapter 6.

3.4.1 Phase One: Research Instrument Development and Data Collection

In this phase, quantitative and qualitative data were collected using four instruments, which include:

- Background Survey.
- Standard English Test (SET).
- Computer Terminology Test (CTT).
- Adjusted Foreign Language Classroom Anxiety Scale (FLCAS).

The collected data also includes the students' academic performance in CS1 courses in the form of quantitative data. Two of the research instruments used were developed as part of this study, including the background survey and the CTT, while the other two, the SET and FLCAS, were adopted from available standard tests. The details of the process of developing and adopting each of those instruments are discussed in the following sections and the summary of research instruments is presented in Table 3.1.

3.4.1.1 Background Survey

A background survey was designed and used to collect both qualitative and quantitative data, featuring closed and open-ended questions divided into six sections. It was made available to students in three languages: Simplified Chinese, Arabic, and English.

The first section aimed to collect demographic data such as name, student ID, and gender. The main aim of collecting personal information is to use it as a key to connect the collected data to the student's participation in the other tests used in this study. For

ethical purposes, the researcher was the only person who had access to this information, and all data were de-identified immediately after the data collection process was completed.

Questions in section two explored aspects related to the student's academic background which may influence their academic performance in programming class. It asked questions about the type of courses the students attended in high school, mainly, science and English subjects. Additionally, a specific question about the medium of instruction used in high school was included. This section aimed to highlight any major differences between the selected case studies that might help explain the research findings.

Section three assessed the students' previous programming experience. As suggested by Byrne and Lyons [70], any study of first-year programming students should consider their previous programming experience. Therefore, this section included questions about formal (courses or classes) and informal (self-education) programming experiences.

Section four explored the language used in the programming class for communication between the students and their teacher as well as for lectures, exams, assignments and learning materials. This type of information could help in understanding and explaining some of the quantitative data analysis results as well as highlighting any differences between the selected cases.

Section five was intended to investigate students' opinions about using English in teaching programming and the difficulties they may face. It contained eight statements derived from the literature about some language difficulties that students could face when using English in class. The students were asked to rate how often they faced each difficulty and the options ranged from '*never*' to '*always*'. An example of these statements is '*I find it difficult to understand some English computer terminology used in the class.*'.

The sixth section of the survey aimed to explore difficulties students faced when learning English-based programming languages such as Java or Python. This section included nine programming activities, and the students were asked to rate how difficult each activity was. The options ranged from '*very difficult*' to '*very easy*' with a '*neutral*' option in the middle. For example, the students were asked how difficult it is to write code using English letters/alphabets and remember the correct spelling of the programming language keywords.

In terms of qualitative data, the last two sections included open-ended questions in which the students could explain or add any challenges they faced in using the English

language in programming class or learning an English-based programming language. Unlike closed-ended questions, open-ended questions aimed to collect genuine opinions from participants without any influence from the researcher [236]. There are many advantages to including open-ended questions in a survey, including the possibility of uncovering unexpected and valuable data, as well as avoiding the bias that may arise from the formatting of the pre-given options in closed-ended questions [237]. On the other hand, the disadvantages of open-ended questions include the amount of time required to code and analyse all responses, as well as the difficulty in collecting the desired number of replies [237]. However, in this research, it was important to collect qualitative data from the students to gain a deeper understanding of the difficulties they face that might not have been addressed in the existing literature. The full background survey is available in Appendix A.

3.4.1.2 Standard English Test (SET)

One of the main aims of this research is to investigate the correlation between NNES students' proficiency in English and their academic performance in CS1 courses. To achieve this aim there was a need to measure the student's proficiency level in the English language. In many previous studies, students' grades in English courses were used to measure their abilities in the English language. However, this method cannot be considered a measuring tool for English language proficiency in general. To overcome this limitation and provide more accurate results, an online standard English test developed by Education First (EF) institution [238] was adopted for this study. Also, this research uses a multiple case study approach, which includes two cases from different institutions. Using the standard English test aimed to improve the comparability between the two case studies.

This test was originally developed by Education First, which is a global English education institution that has more than 600 offices and schools around the world. The test has been in use since 2014, and its results show high reliability [238]. The test can be completed in 50 minutes, and the outcome is a score ranging from 1 to 100. These scores are mapped onto six language ability levels: '*beginner*', '*elementary*', '*intermediate*', '*upper-intermediate*', '*advanced*', and '*proficient*' [239]. The arrangement to make this test available to all participants in this study was made in collaboration with the EF management team in Saudi Arabia.

3.4.1.3 Computer Terminology Test (CTT)

To focus this investigation more directly on the English language skills of CS1 students and to explore their knowledge of English computer terminology, a special test was developed and validated as part of this thesis. The first step in the process of designing this test was to compile a CS1 word list by analysing four CS1 textbooks and lecture notes from nine courses. The second step was to decide on the words to be used in the test as using the full CS1 word list (123 words) seems unpractical. Instead of randomly selecting a number of terms to build the test, the produced word list was shared with four teachers experienced in teaching programming to NNES students. The teachers were asked to rate the words from most important to least important. Then, the 30 most important words based on the teachers' voting were used in building a multiple-choice test that is referred to in this thesis as the Computer Terminology Test (CTT).

In the following step, a short and simplified definition was prepared based on a combination of online resources (open-source educational websites), books and dictionaries for each of the 30 top-rated words. In the test, the students were asked to select the correct term out of three options for the provided definition. The test draft was reviewed by two CS1 teachers to ensure the clarity and accuracy of questions and the provided answers. The design of the test considered the optimal number of answer options for each question to be three as suggested by Rodriguez [240]. To achieve a more accurate result the answer options for each question were in a random order [241]. The CTT was provided online for all participants with a time limit of 30 minutes to take the test, thereby minimising the likelihood of cheating. Finally, to verify the reliability of this test a validation analysis was conducted using the data collected from 150 participants from the two groups. More details about the processes of generating the CS1 word list as well as building and validating the CTT are presented in detail in the next chapter (chapter 4) and the full CTT is provided in Appendix B.

3.4.1.4 Adjusted Foreign Language Classroom Anxiety Scale (FLCAS)

Horwitz et al. [142] developed what they called the Foreign Language Classroom Anxiety Scale (FLCAS) for testing the level of students' anxiety in foreign language classes. This test originally included a 33-item, five-point Likert scale in which the students were asked to choose one of the following options: *'strongly agree, agree, neither agree nor disagree, disagree, or strongly disagree,'* to answer each item. For each item, a score was given ranging from 1 for *'strongly disagree'* to 5 for *'strongly agree'*. To analyse the results correctly, the negatively worded items were assigned a reversed order of scoring so that a higher score would indicate a higher anxiety level [242]. The minimum

possible score is 33 which represents the lowest level of FLCA whereas the maximum possible score is 165 which represents the highest level of FLCA. Since the FLCAS was published, it has been used in different studies across several disciplines and languages and has proven to be a very reliable tool [242, 243, 244].

According to Horwitz et al. [142], this scale covers three types of anxieties related to foreign language anxiety including; communication apprehension, fear of negative evaluation, and test anxiety. However, the scale developers did not provide a clear and specific explanation of the underlying components of the scale [242, 245]. In other words, the FLCAS measures the general level of foreign language anxiety students experience in the class but does not explain which items in the scale can be used to measure each factor or component.

Many researchers tried to identify the underlying components of the FLCA scale developed by Horwitz et al. (e.g. [242, 246, 243, 247, 248, 249, 250, 245]). As a result, a number of different factors were identified including speech anxiety, fear of negative evaluation, fear of failing the class, comfortableness in speaking with native speakers, low self-confidence in speaking English, fear of not understanding and communication apprehension. Each of those factors can be measured using some of the original FLCAS items. However, this thesis focuses on the overall level of FLCA that some students face in CS1 courses. Thus, analysing the underlying factors of the scale is not part of this thesis even though it was investigated and presented in a separate conference paper [251].

In this research, the FLCAS was slightly adjusted to test the students' level of anxiety when using English in CS1 class (Appendix C). For example, the original phrasing of item 2 in the FLCAS is '*I don't worry about making mistakes in English class*'; the rephrased item 2 that was used in this research is '*I don't worry about making language mistakes when I am speaking English in class*'. The adjusted FLCAS was made available to participants in their native languages, alongside English, to avoid any misunderstanding or bias during data collection.

Cronbach's Alpha was calculated to ensure that the adjustment and translation did not affect the instrument's validity. Cronbach's Alpha is usually used to measure the internal consistency reliability of multiple-item measure instruments [252]. Thus, it was used to ensure the reliability of FLACS in research when the original questionnaire was slightly adjusted as in [253, 42]. In this research, Cronbach's Alpha was calculated at $\alpha = 0.96$ based on data collected from 160 participants from both groups. This result indicates that the adjustments made to the scale did not impact its reliability.

Research Instrument	Source	Aim	Type of data collected
Background Survey	Developed by the researcher.	- Demographic data, - Academic background, - Previous programming experience, - Language used in the CS1 class, - Students' perspectives about using English in CS1 courses, - Students' perspectives about studying English-based programming languages.	Quantitative & Qualitative
Standard English Test (SET)	EF institution [238].	Measuring students' proficiency level in English.	Quantitative
Computer Terminology Test (CTT)	Developed and validated by the researcher [254, 255].	Measuring students' knowledge of CS terminology used in CS1 courses.	Quantitative
Adjusted Foreign Language Classroom Anxiety Scale (FLCAS)	FLCAS by Horwitz et al. [142].	Measuring the level of FLCA students' experience in EMI CS1 courses.	Quantitative

Table 3.1: Summary of research instruments.

3.4.1.5 Data Collection Process

The data used in the multiple case study were collected online over two semesters in two consecutive academic years (20/21 - 21/22). The data collected to evaluate the managed and peer-reviewed translation activity was collected in the first semester of the academic year of 22/23. The main reason for collecting all data through online means was the COVID-19 pandemic and the lockdown that affected all the educational institutions included in this research.

In both institutions, each semester is between 14 to 15 weeks long and starts about the same time of the year, which in this study was the semester starts in September (first semester). The first data collected in this research was English language proficiency using the SET. The test was available to students between weeks two and four of the semester. After that, the CTT was available for two weeks followed by two weeks for the adjusted FLCAS in the middle of the semester. The last instrument used was the background survey which was made available for three weeks at the end of the

semester. Scheduling the release of tests and surveys considered the different holidays that each country has throughout the semester. The final academic performance of the students in CS1 courses was collected three to four weeks after the end of the semester. The timeline for data collection shown in Figure 3.1 illustrates the order of using research instruments for data collection throughout the semester. The same timeline was used to collect the multiple case study data for both academic years. More details about the data collection process for the evaluation of the developed translation activity, which was completed in another academic year, are presented in chapter 6.

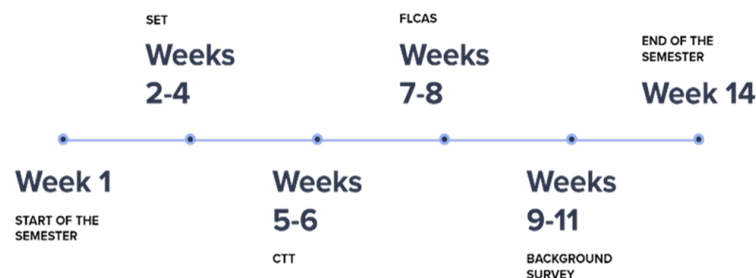


Figure 3.1: Data Collection Timeline

3.4.2 Phase Two: Data Analysis

Data analysis in this thesis has three main aims; the first is to examine the validity of some research instruments and the data they produced, especially the newly developed CTT. The second aim is to test the research hypotheses and answer the research questions, while the third aim is to prepare for the third phase, which is the cross-case analysis.

In this phase, the primary data generated from phase one was reviewed, cleaned, coded, verified, and de-identified in preparation for analysis. Data cleaning involves dealing with incomplete, duplication, and errors in the data. After that, the data needed to be coded or represented in a form that would allow statistical analysis to be performed. It is also important to mention that all data have been de-identified and all personal information deleted after matching the records from different datasets. Data collected from each semester were de-identified immediately after the data collection process was completed. De-identification of personal information is discussed in the ethical consideration section 3.6.

In this research, the instruments adopted are separate, so the data cleaning and preparation were done in overlap with the collection of other data for both case studies (Figure 3.2). For example, the English test results have been cleaned, coded and veri-

fied at the same time CTT and FLCAS were being collected. This approach helped in verifying each instrument and the data it produced before the data collection period ended to avoid any errors.

Moreover, the data collected from the first semester were used to validate the CTT and the adjusted FLCAS. This process ensured the suitability of the instruments and increased confidence in using them to collect the second set of data in the following year. Another benefit of this approach is that it helps overcome the problem of the large amounts of data that can be generated in case study research. Applying this tactic of overlapping and iterative analysis reduced the pressure of having a huge amount of primary and unprocessed data at the end of the data collection phase, which is one of the drawbacks of the case study approach.

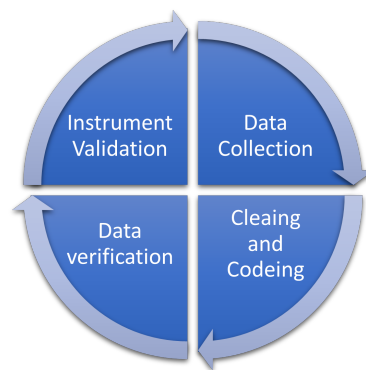


Figure 3.2: Data preparation and instruments validation process

Once the data was prepared, different statistical techniques were applied using IBM SPSS statistics software [256]. The first stage of analysis aimed to validate the data and provide a general overview of it. In this step, some basic descriptive analyses were performed, including the use of standard deviations and mean values (averages). Additionally, the minimum and maximum values for each variable were checked, when possible, to ensure that all values were within acceptable levels. Instruments validation involved conducting more specific types of analysis that ensure the validity and reliability of those instruments. For FLCAS, Cronbach's Alpha was used for validation as provided in section 3.4.1.4 and for the validation process of CTT see Chapter 4: section 4.3.

The next, and more crucial, step was to explore the relationship between the selected factors and students' academic performance in CS1 courses. In statistics, correlation analysis is a method to find associations or relationships between two or more numerical variables [257]. Pearson's correlation coefficient is usually used when the data is assumed to have a linear relationship, the variables are independent of each other and normally distributed [257]. These assumptions were checked, and the test appeared to

be suitable for the collected data. Moreover, previous studies in the area have used this statistical test to investigate the correlations between different factors and students' academic performance (e.g. [58, 66, 72, 55, 59, 29, 70]). Therefore, Pearson's correlation coefficient was used to test the correlations between the dependent and independent variables as provided in Table 3.2.

The last important step involved using multiple linear regression analysis to identify the ability of the studied factors to predict NNES students' academic performance in CS1 courses. The main idea of the linear regression technique is to analyse data to build a prediction model [258]. There are two types of linear regression, simple and multiple. In simple linear regression, a model is built to predict a response variable (y) from an explanatory variable (x) whereas, in multiple linear regression, several explanatory variables are included ($x_1, x_2, \dots, x_p$). The equation for multiple linear regression is:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi} + e_i \quad (3.1)$$

Moreover, the answers to some of the research questions lie in the data collected through the background survey. Also, it contains useful data such as demographical data, academic background, and previous programming experience, as well as open-ended questions. Therefore, comprehensive quantitative and qualitative analyses have been conducted on the data collected through this instrument, which added valuable information to the research. It also helped in highlighting some of the differences and similarities of the case studies, which explains some of the findings.

The responses to open-ended questions that appeared in the background survey were analysed using the inductive coding method based on the exploratory nature of these questions. This method allows the researcher to read and interpret the raw data based on context to generate suitable themes that fit the data at hand [259]. After generating the main themes, all responses should be categorised based on those themes; however, during this process, new themes might appear or multiple themes might be merged into one main theme [259]. The final set of themes is one where each response can be placed under a single theme. Nevertheless, some responses might contain more than one idea; therefore, they can be placed under multiple themes. Table 3.2 provides a summary of the statistical analysis techniques that were used to answer each of the research questions.

Research Question	Research method	Independent Variable\s	Dependent Variable\s	Statistical Test/s
Q1.1: What is the influence of NNES students' proficiency in English on their academic performance in CS1?	- SET, - CS1 academic performance.	English proficiency level	- Assignment Score, - Final exam Score, - Overall CS1 score.	Pearson coefficient correlation
Q1.2: What is the influence of NNES students' knowledge of English computer terminology on their academic performance in CS1?	- CTT, - CS1 academic performance.	Knowledge level of computer terminology	- Assignment Score, - Final exam Score, - Overall CS1 score.	Pearson coefficient correlation
Q1.3: What is the influence of foreign language classroom anxiety on NNES students' academic performance in CS1?	- Adjusted FLCAS, - CS1 academic performance.	Level of Foreign Language Classroom Anxiety	- Assignment Score, - Final exam Score, - Overall CS1 score.	Pearson coefficient correlation
Q2: To what extent can online managed and peer-reviewed translation activity for computer terminology improve students' knowledge of terminology related to the CS1 course they are attending? ¹	- CTT, - Feedback Survey.	-	-	- Descriptive statistics, - Two-sample t-test. - Inductive coding for the open-ended questions.
Q3: What are the perspectives of NNES students about using the English language in CS1 courses?	- Background Survey.	-	-	- Descriptive statistics, - Inductive coding for qualitative data.

Table 3.2: Summary of the statistical analysis techniques.

¹ The details of the methodology used to address this research question are presented in Chapter 6

3.4.3 Phase Three: Cross-Case Analysis and Hypotheses Testing

The cross-case analysis, which is sometimes called a comparative case study, is the analysis process that seeks to identify similarities and differences between multiple cases in case study research [260]. The results can be more compelling when more cases with a greater variety are included in this analysis [233]. In cross-case analysis, a larger scope is considered when analysing and interpreting the results, such as two or more classrooms or schools [260].

The generalisability of the research findings increases if similar patterns are detected across cases during the cross-case analysis. However, the main aim of the cross-case analysis is transferability, not generalisability, of the findings. Transferability encourages the reader of the research to find connections between the study's findings and their own experience, whereas generalisability involves extending the research findings to include other similar cases not included in the study [261].

The cross-case analysis in this research included a comparison of the participants' academic background and their perspectives on using English in CS1 classes and studying English-based programming languages. The results highlighted both similarities and differences between the two cases. The figures from the correlation analysis were also compared, and further analysis was conducted to determine whether the correlation results of the two cases were statistically different from each other. The correlations were compared using Fisher's r -to- z transformation analysis (Equation 3.2) [262]

$$z = \frac{1}{2} \log\left(\frac{1+r}{1-r}\right) \quad (3.2)$$

This analysis is usually used to test the significance of the difference between two correlation coefficients, r_1 and r_2 from two independent samples. The z -value will be positive if $r_1 > r_2$ and negative if $r_1 < r_2$. However, the difference will be only significant if the p -value of the z -value is < 0.05 , otherwise, it can be said that there is no significant difference.

The cross-case analysis also compared the results of multiple linear regression which has been used to find the best predictor(s) for the NNES students' academic performance in CS1 courses out of the three studied factors. In addition, the result of the multiple case study was used to test the research hypotheses, particularly the multiple linear regression results [263].

3.5 Participants and Sample Size

Based on the case study boundaries, the participants in the two cases were first-year university-level NNES students who were enrolled in CS1 courses (the first programming course in their programme). In Case Study One (China), the students were computer science and engineering majors, while in Case Study Two (Saudi Arabia), they were all computer science students. However, they were all attending English language courses simultaneously with their CS1 courses as part of the required courses for their programme. 'Case Study One' included participants from Beijing-Dublin International College (BDIC) in China while 'Case Study Two' included participants from Shaqra University (SU) in Saudi Arabia.

The countries where those institutions are based are non-English speaking countries and they have completely different native languages. However, in both institutions, the medium of instruction, particularly in CS1 courses, is English. This means that all lectures, exams, and assessments are provided in English. The main difference was that the instructors in Case Study One (China) were native English speakers and did not speak the students' native language, whereas the instructors in Case Study Two (Saudi Arabia) spoke English as a second language and shared the same native language as the students.

The aim of including two case studies with participants from different backgrounds, institutions, and countries is to obtain a deeper understanding of the studied topic. Moreover, as explained in the methodology, the multiple case study can give more generalisable and transferable results which makes the research more robust. This research is one of the few studies in the field that has been conducted in more than one institution and across more than one country [264].

In general, sample sizes between 30 and 500 are considered appropriate for most types of research, but if the sample is to be analysed as subsamples, each should consist of no fewer than 30 participants [265]. In this research, the total sample size ranges from 303 to 393 for each of the research instruments. The case studies were analysed separately, and the sample size for each instrument in each case study is greater than 100, as shown in Table 3.3.

In research that investigates more than one variable and conducts analysis such as multiple regression, the sample size should be at least 10 times larger than the number of variables [265]. The multiple regression analysis in this research included 3 independent variables and one dependent variable, meaning the sample size should be greater than 40. In fact, the sample size used in the multiple regression analysis was between 20 and 30 times larger than the number of variables.

Moreover, the minimum generalisable sample size for prediction models for a large population as defined in a similar study is 96, with a $\pm 10\%$ acceptable margin of error [266, 264]. The sample size in this study is over 96, and as is known the margin of error decreases as the sample size increases [266]. Based on that, it is possible to say that the sample size of this research can be considered highly acceptable and the results can be highly reliable. It is important to mention again that this research does not aim to build a prediction model but uses multiple linear regression to test the ability of the studied factors to predict students' academic performance.

The detailed sample sizes for the two case studies are provided in Table 3.3. The number of students attending the CS1 course at SU, Saudi Arabia (Case Study Two), was significantly lower than the number of students attending similar courses at BDIC, China (Case Study One), which explains the difference in the sample sizes. More details about the participants, such as gender, age, and background, are presented in chapter 5.

	Case Study One (China)	Case Study Two (Saudi Arabia)	Total Participants
Background Survey	276	117	393
SET	190	113	303
CTT	184	117	301
FLCAS	239	114	353

Table 3.3: Number of participants in the multiple case study

3.6 Ethical Considerations

As part of preparing for data collection, appropriate ethical approvals were obtained from the ethical committees at both institutions. The research received ethical exemptions with the reference "*LS-E-20-129-Alaofi-Russell*" from University College Dublin for Case Study One and ethical approval with the reference "*Ethics Appl.202001*" from Shaqra University for Case Study Two.

All required measures for data security, protection, and privacy were implemented during and after the data collection process. Personal information has been de-identified after matching datasets from different research instruments in a way that makes it no longer traceable. A common cryptographic function called MD5 (Message Digest Method 5) was used in this research to de-identify the data. The use of this tech-

nique enabled the de-identification of the data without losing the ability to match the data of corresponding records that are collected from different sources.

The collected data was stored on a personal computer used only by the researcher and protected with a password and security software. The online cloud storage used for backup is one of the most secure platforms, and the files were secured with a password before being uploaded to the cloud for an extra layer of protection.

Moreover, participants were provided with an information sheet that explained the research aim, objectives, and all other relevant details regarding their participation in the study. Although participation in the research was completely voluntary, informed consent was obtained from participants before data collection. The consent form included the researcher's contact email in case the participants had any further questions.

Another important ethical consideration, especially in multilingual research, is the language used to communicate with participants. It was very important to ensure that participants fully understood the informed consent before agreeing to participate. For this reason, the informed consent and the research information sheet were made available in three languages: Simplified Chinese, Arabic, and English.

Additionally, the background survey, FLCAS, and the feedback survey on the translation activity were all provided in the students' native languages, as well as in English. In addition to ethical reasons, this step is crucial to eliminate any potential bias related to providing a survey only in English when the target audience consists of NNES individuals. In a similar study, Guo [10] states that using English in their survey could have negatively impacted their research results because only people proficient in English would complete the survey.

In this research, participants were able to choose any of the three languages to complete the surveys, which can help provide more reliable data [267]. This was also expected to increase the participation rate and enable students to provide more informative answers, especially to open-ended questions, and prevent any misunderstanding of the questions. The translations were conducted by translation specialists and reviewed by native speakers of both languages.

3.7 Conclusion

This chapter provided an overview of the research framework, methodology, methods, and design. The main aim of this chapter was to describe and justify the methods used, as well as to provide details on the process of data collection and analysis. The other

important aim is to provide enough information for future researchers who could be interested in replicating this research or some parts of it. The detailed description of the research instruments can help researchers adopt them in their future research.

The next chapter describes the process of developing and validating the CTT, one of the primary research tools. A separate chapter has been dedicated to this instrument due to its significance and the many stages involved in its production. Additionally, as it is a newly developed research instrument, it needed to be tested and validated. The CTT is one of the main contributions of this thesis; therefore, its development and validation process had to be explained in great detail.

COMPUTER TERMINOLOGY TEST: DEVELOPMENT AND VALIDATION

4.1 Introduction

One of the factors that this thesis aims to investigate is students' knowledge of computer terminology used in CS1 courses and its impact on their academic performance. However, a standard test for this purpose does not appear to be available in the literature. This chapter presents the process of developing and validating a Computer Terminology Test (CTT) specifically designed for the CS1 course. The aim of this test is to measure students' knowledge of the most common CS terminology used in CS1 courses.

The CS1 word list was generated as the first step in building the terminology test. Following this, the most important words were identified with the help of four experienced CS1 teachers. The result was a sub-list of 30 words, which were used to construct the CTT in the form of a multiple-choice test. Finally, the test was validated using the Rasch model analysis approach to analyse students' responses during the first phase of data collection ¹

¹Large parts of this chapter were published and presented in the following work:

- Alaofi, S. and Russell, S. '**Computer Terminology Test for Non-native English Speaking CS1 Students.**' in *In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE)*, poster, 2021, 1 page. [Online].
- Alaofi, S. and Russell, S. '**A Validated Computer Terminology Test for Predicting Non-native English-speaking CS1 Students' Academic Performance.**' in *Proceedings of the Australasian Computing Education Conference (ACE)*, article, 2022, 10 pages. [Online].

4.2 CS1 Word List

4.2.1 Background

The meaning of specific English terminology can vary depending on the field in which it is used. Field-specific word lists offer many benefits, one of which is helping NNES students in their learning through EMI [93]. Many researchers have dedicated part of their work to generating technical word lists in different fields. Examples include the Academic Word List (AWL) [268], the Medical Academic Word List (MAWL) [269], the Basic Engineering List (BEL) [270], and the Business Word List (BWL) [271]. In CS, four available word lists and supplementary lists have been generated from different types of resources and cover various CS sub-disciplines [93, 206, 272, 273].

The Computer Science Word List (CSWL) was the first CS-specialised word list, generated by Minshall [93]. This list was compiled mainly from journal articles and conference proceedings and covers 10 CS sub-disciplines. These include computer systems organisation, computing methodologies, hardware, human-centred computing, information systems, mathematics of computing, networks, security and privacy, software and its engineering, and theory of computation. The final list contained 433 single-word terms and 23 multi-word terms. The CSWL was designed for university-level students across different computer science disciplines. However, its dependence on journal articles and conference papers, rather than textbooks or lecture notes, may be viewed as a limitation. Additionally, the list is quite long for first-year undergraduate students and includes terms that they may not need at this level.

Another word list was developed by Bi [206], and it was the first to focus on the CS undergraduate level. The Computer Science Vocabulary List (CSVL) was compiled based on undergraduate textbooks covering various CS subjects. The selected textbooks included topics such as computer architecture, computer networks, operating systems, data structures, algorithms, software engineering, information security, cryptography, computer graphics, artificial intelligence, and different programming languages. The final CSVL contains 356 single-word terms but does not include any multi-word terms. Unlike CSWL, CSVL was specifically designed for undergraduate students and was based primarily on learning materials. However, the topics covered in this list are not necessarily related to programming, particularly at the beginner level. Furthermore, CSVL was mainly intended for a culture-specific audience, as it excludes English academic words that Chinese undergraduates were expected to have mastered before university.

The Computer Science Academic Vocabulary List (CSAVL), developed by Roesler [272], is intended to support CS students at different educational levels in their academic studies. The main word list consists of 904 words designed to help students achieve a minimal level of academic reading comprehension in the CS field. Additionally, the researcher provided a supplementary word list (CSAVL-S), which includes another 702 words related to CS. The sources used to compile both the main and supplementary lists were journal articles and textbooks covering 10 different CS sub-disciplines. Similar to the work of Bi [206], the researcher did not include any multi-word terms in either list. Furthermore, the topics covered in these lists extend far beyond introductory programming courses, focusing more on academic reading comprehension rather than programming-related content. Both the main and supplementary lists are extensive and may be impractical for students or teachers in CS1 courses, as they contain many terms that are not relevant to that course.

Uba et al. [273] recently published another CS word list aimed at supplementing the work of Minshall [93] and Roesler [272]. This word list, called the New Computer Science Academic Word List (NCSAWL), contains 444 words covering 10 CS sub-disciplines. The researchers argued that NCSAWL includes words that are more frequently used in CS rather than general academic English, making it more specialised than CSAVL by Roesler [272]. As with all previously discussed word lists, NCSAWL covers a much broader scope than the CS1 course. Moreover, like CSVL and CSAVL, this list does not include any multi-word CS terms.

Although CSWL, CSVL, CSAVL, and NCSAWL are CS-specialised word lists, they contain a mix of words from various topics and sub-disciplines within the CS field. Each of these lists includes hundreds of words, with CSWL being the only one to include multi-word terminology in addition to single-word terms. The overlap between CSWL and CSVL, as calculated by Bi [206], is 144 words (40.45%). The low overlap was attributed to differences in the materials used, the purpose of creating the lists, and the target audience. However, the overlap between CSWL and CSAVL, including its supplementary list (CSAVL-S), was even lower at 23% and 29% respectively, as calculated by Roesler [272]. The same factors explained by Bi [206] may have also contributed to this low overlap.

Three main reasons make the available CS word lists unsuitable for building the CTT for CS1 students. Firstly, these lists cover topics beyond what is needed for CS1 courses. Secondly, they are based on materials that either contain professional and academic language, such as journal articles and conference papers, or textbooks intended for more advanced undergraduate levels. Thirdly, the target audience for these lists is not first-year students in CS1 courses. Therefore, there was a need to develop a specialised

word list for CS1 courses. Besides its importance for building the CTT, this list will be highly beneficial for NNES students learning CS1 through EMI. Additionally, the CS1 word list can help shape and refine the English language courses provided to NNES students in the CS major.

4.2.2 Methods and Tools

The first step was to build and prepare the corpus used to generate the word list. This required selecting appropriate materials that aligned with the word list's aim and audience. The goal was to generate a CS1 word list suitable for CS1 or any introductory programming course and for students attending these courses, regardless of their level of English proficiency. Therefore, the first criterion for selecting materials was to include only CS1 or introductory-level programming textbooks. Secondly, the selected textbooks covered four of the most popular programming languages: C, C++, Java, and Python. This diversity in programming languages aimed to make the list applicable to different CS1 courses. Moreover, the chosen textbooks and lecture notes were all open-source to avoid any copyright violations. Lastly, all textbooks and lecture notes had to be in PDF format for analysis purposes. Based on these criteria, four textbooks and nine full-course lecture notes were selected to build the corpus.

After selecting the appropriate materials, the first step was to prepare them for analysis. In this stage, all selected files were converted into plain text and organised in a way that allowed them to be analysed by specialised software in the next step. A total of 1.3 million tokens (words) were extracted and prepared for analysis. The software used for format conversion and word extraction was AntFileConverter [274], an open-source program. The next step involved analysing the generated text files to determine the frequency and range of each word. Two types of words were included in the analysis: single-word terms (e.g. '*algorithm*') and multi-word terms (e.g. '*operating system*'). Two different software programs were used for each type of word: AntWordProfiler [275] for single words and AntConc [276] for multi-word terms. Both are open-source programs.

The *range* refers to the total number of sources in which a word appears, while *frequency* denotes the number of times the word occurs across all sources [93]. For a word to be included in the word list, it must appear in at least half of the selected sources [268, 93, 206]. Therefore, for the CS1 word list, each word must appear in at least two of the textbooks or five of the lecture notes.

On the other hand, there is no specific rule to determine the acceptable frequency, nor are there explanations regarding the rationale behind the values used in previ-

ous studies [206]. Many word list studies, including CSWL [93], followed Coxhead [268]’s method and set the minimum frequency at 28 per million words. However, Bi [206] adopted a slightly different approach, commonly found in other research, when generating the CSVL. They examined several frequency levels before determining the optimal threshold for their selected sources, ultimately setting it at 13.31 per million words. Similarly, in this thesis, various frequency levels (ranging from 10 to 40 per million words) were tested. The frequency threshold found to be the most suitable for the size of the analysed sources was 40 occurrences (30 per million words). Figure 4.1 summarises the steps involved in generating the CS1 word list.

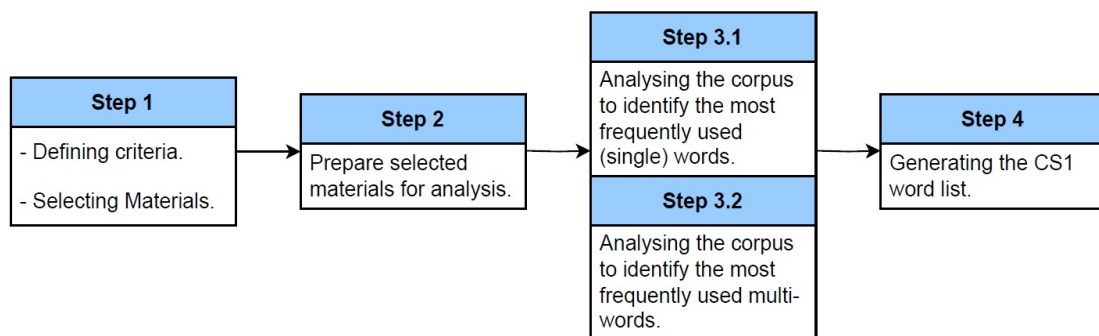


Figure 4.1: The steps of generating the CS1 word list

4.2.3 Results

The final result was a list of 123 of the most frequently used words and multi-word terms in CS1 courses: 110 single words and 13 multi-word terms (Table 4.1). It is a relatively short list compared to previous CS word lists; however, it specifically covers only one sub-discipline of CS-programming, particularly at the introductory level. Nevertheless, some overlap is expected between the newly generated CS1 word list and previous CS word lists, as they encompass multiple CS sub-disciplines, including programming-related topics.

To evaluate a word list, it is always recommended to compare it with other word lists that share the same purpose [277]. NCSAWL [273] was not included in the comparison as it was published in 2023, two years after generating the CS1 word list and the CTT, and it aims to complement other word lists. Nonetheless, it has been verified that NCSAWL does not contain any of the words that appeared in the CS1 word list but were not covered by the other word lists.

Table 4.2 shows the number of shared and unshared words between the CS1 word list and previous word lists in the field of CS. The result of the comparison showed that

algorithm	database	hardware	iterate	prefix	subtract
alphabet	debug	header	keyboard	printer	swap
arithmetic	decimal	height	keyword	processor	switch
array	decrement	hexadecimal	linear	programmer	syntax
binary	default	hint	literal	prototype	tab
blank	delete	horizontal	loop	pseudo	template
boolean	digit	identifier	lowercase	recursive	terminology
bracket	directory	import	mathematics	reference	token
bug	disk	increment	matrix	relational	unsigned
byte	duplicate	indent	merge	reset	update
cast	electronic	inherit	minus	reuse	uppercase
client	embed	initialize	module	runtime	usage
column	execute	integer	negation	semantic	variables
comma	exit	interface	notation	semicolon	vertical
compiler	exponent	internet	null	server	web
concatenate	factorial	interpreter	numeric	software	
console	function	intersection	operand	stack	
constructor	garbage	invalid	overview	static	
counter	graphic	inverse	parentheses	storage	
binary tree	control- structure	data type	global- variable	operating- system	user- interface
central- processing-unit (CPU)	data- structure	execution- time	local- variable	software- development	virtual- machine
control-flow					

Table 4.1: CS1 Word List

there are 60 single words and seven multi-words shared between CSWL and the CS1 word list (total overlap 54.5%). This means 56 words appearing in the CS1 word list are not covered by CSWL, and a large number of words (386) in that list might not be related to CS1 topics.

Comparing the CS1 word list with CSVL reveals that there are 58 words shared with the CSVL (47.2%). Again, 65 words that appeared in the CS1 word list are not included in CSVL, and the rest of this list, which contains 298 words, might not be related to the CS1 course.

The overlap between the CS1 word list and CSAVL and its supplementary list is 56 words (45.5%) and 36 words (29.2%), respectively, which leaves 31 words of the CS1 word list not covered by the two lists combined. The remaining 1514 words in the

Word List (<i>n</i> of words)	<i>n</i> and % of words appeared in CS1 word list	<i>n</i> and % of words in CS1 word list NOT covered	<i>n</i> and % of words NOT in CS1 word list
CSWL (456)	67 (54.5%)	56 (45.5%)	386 (85.3%)
CSVL (356)	58 (47.2%)	65 (52.8%)	298 (83.7%)
CSAWL (904)	56 (45.5%)	67 (54.5%)	848 (93.8%)
CSAWL-S (702)	36 (29.2%)	87 (70.7%)	666 (94.9%)

Table 4.2: Matching percentage between the CS1 word list and previous word lists

two lists might not be needed for the introductory programming level, such as the CS1 course.

A total of 14 terms from the CS1 word list, which represent about 11.4%, have not been covered by any of the previous word lists, including the most recently published list (NCSAWL). Figure 4.2 is an illustration of the overlap between the CS1 word list and the previous word lists in the field, which shows that large parts of those word lists might not be related to introductory programming. This is expected because all previous lists cover up to 10 sub-disciplines of CS, where programming is only one of them.

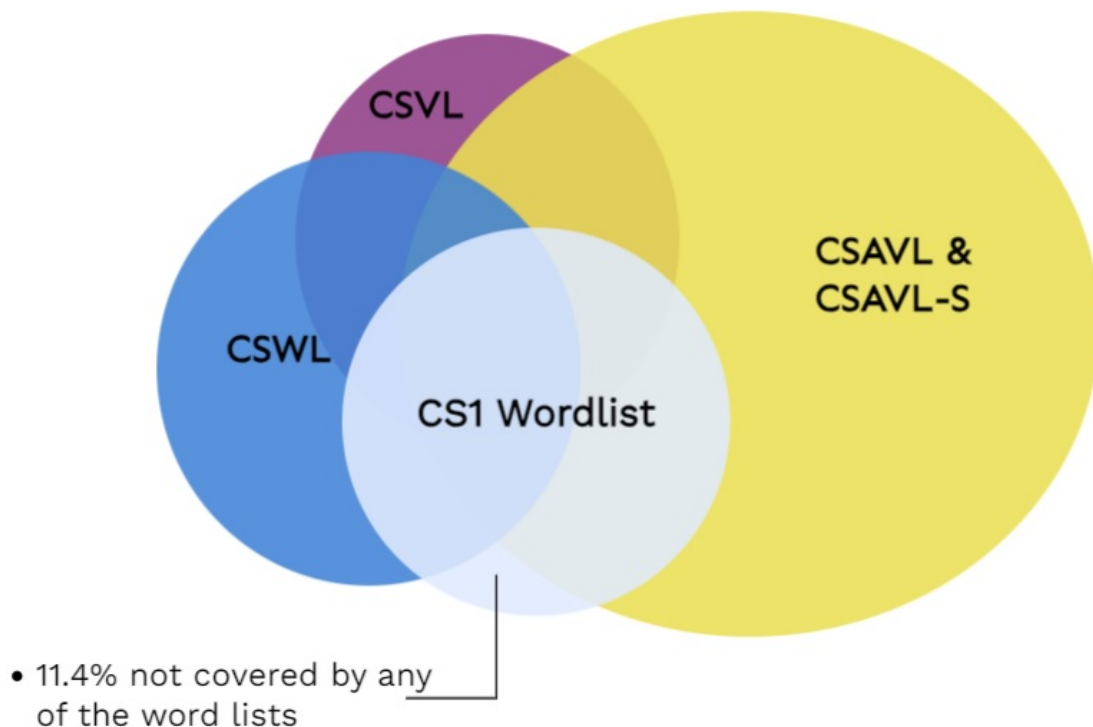


Figure 4.2: The overlap between previous word lists and the CS1 word list

The overlap between word lists is not limited to specialised lists from the same field. For example, CSVL showed a 27.5% overlap with the Academic Word List (AWL) [268], which is a more general-purpose word list [206]. Therefore, it is fair to say that overlap between word lists is always expected, and this overlap can be higher if they are related to the same field. However, this does not reduce the importance of creating a specialised word list, which can be more focused and target a particular audience.

The main reason for compiling the CS1 word list in this thesis was to build a specialised computer terminology test targeting CS1 students in particular. This could not be achieved with word lists that include terms from different subjects, even if they are still CS-related. The other expected benefit of this list is to provide NNES students in CS1 courses with a word list that can help them in their CS1 course with minimal confusion. While the previous CS word lists can be very helpful, the topics they cover and the levels they target might overwhelm beginners with words they do not actually need at this stage. These word lists are quite long compared to the new and more focused CS1 word list. Moreover, there are eight single words and six multi-word terms in the CS1 word list that did not appear in any of the previous lists. Those words are: *blank, cast, comma, console, indent, interpreter, minus, semicolon, binary tree, central processing unit, control structure, data type, global variable, and local variable*.

Finally, to confirm the relevance of the produced CS1 word list, it is important to present the results of a study conducted by Becker et al. [278], where this list was used. In their study, which focused on the speech used by teachers in programming classes, they analysed audio transcripts of three introductory programming courses. The courses were taught by different teachers across different institutions located on three continents. The researchers calculated the percentage of words used that appeared in the Free Online Dictionary of Computing (FODOC) and the CS1 word list. The results showed that the use of CS1-specific words remained extremely consistent in the three courses. The number of times that words from the CS1 word list were used by the teachers ranged from 938 to 2284 in each of the analysed courses. This means that the CS1 word list was used between 762.6% and 1856.9% during the CS1 courses compared to its size, which is only 123 words.

4.3 Computer Terminology Test (CTT)

4.3.1 Development Process

There is no agreement in the literature on the sufficient number of questions when designing a multiple-choice test. Some research considers 60 test items in one hour of exam time, following the rule of one item or question per minute, as the optimal design [279]. Another suggests that a 40-item test is more than adequate if content and validation constraints are met [280]. However, many researchers consider 30 items to be sufficient to yield reliable results, as excessively long tests can overload both students and teachers [281, 279]. Thus, using all 123 words to build a single multiple-choice test seemed impractical, especially since it would be introduced to students alongside other tests and surveys. The students would be asked to participate in this test voluntarily along with all other research instruments, so it should take a reasonable amount of time to complete. A long test might also negatively impact the participation rate.

Therefore, the CS1 word list was shared with four CS1 teachers who had at least two years of experience teaching CS1 courses to NNES students. The aim was to extract the most important words from the list and use them to build the multiple-choice test. Using an online form, the teachers were asked to rate each word in the CS1 list from most to least important. The top words were those rated as either '*very important*' or '*important*' by all four teachers. Any word that received one or more ratings as '*not important*' was excluded.

The result was a sub-list of 30 words that were perceived as the most important for CS1 courses out of the 123-word list (Table 4.3). Consequently, the decision was made to limit the number of CTT questions to 30 multiple-choice questions using the most important words in the CS1 word list.

algorithm	array/list	binary	boolean	comma
compiler	control structure	counter	data structure	data type
debugging	execution	exit	function	identifier
import/include	increment	indentation	initialization	integer
interpreter	iteration	keywords	loop	module
numeric	parentheses	processer	syntax	variables

Table 4.3: The 30 most important terms in the CS1 word list

In the process of building the CTT, accurate and concise definitions of each term in the sub-list were selected from various sources, including open-source educational websites, books, and dictionaries (e.g. [282, 283]). The design of the test considered the optimal number of answer options for each question to be three, including two distractors (wrong answers) and one correct answer [240, 284]. To achieve more accurate results, the answer options for each question were presented in random order [241]. The test draft was reviewed by two CS1 teachers to ensure the clarity and accuracy of the questions and answer options. The test was then designed using an online platform that allows for setting a time limit. The time limit was intended to provide a fair and equal chance for all students while minimising the likelihood of cheating. Following the one minute per item rule [285], the test time limit was set to 30 minutes.

4.3.2 Test Validation Background

A test can be considered valid if it measures what it is intended to measure [286]. A more specific definition of test validity, as given by Messick [287], is 'an overall evaluative judgment of the degree to which empirical evidence and theoretical rationales support the adequacy and appropriateness of interpretations and actions based on test scores or other modes of assessment'. Six aspects of test validity were identified by Messick [287, 288], including content, substantive, structural, generalisability, external, and consequential. The following is a brief description of each of these six aspects:

- **Content aspect:** Concerned with the relevance and representativeness of the test content to the construct domain and its technical quality.
- **Substantive aspect:** Relates to finding evidence that test takers are genuinely engaged with the tasks included in the test.
- **Structural aspect:** Deals with the scoring scheme.
- **Generalisability aspect:** Concerned with the extent to which the meanings and interpretations of the scores can be generalised to other tasks that are not included in the test but are part of the domain under examination.
- **External aspect:** Concerned with the relationships between the test scores and other test and non-test behaviours.
- **Consequential aspect:** Concerned with the actual and potential consequences of score interpretation on future actions.

Smith Jr. [289] presented the possibility of using the mathematical Rasch model [290] to find empirical evidence for the validity described by Messick. Following Smith Jr.'s work, several researchers contributed to establishing criteria for determining each aspect of validity using this mathematical approach [291]. Moreover, various software programs, such as Winsteps and Facets, have been developed to apply the Rasch model [292], making it more accessible to researchers from different backgrounds.

The Rasch model provides several analyses that can be used to find empirical evidence for each aspect of construct validity [289, 293, 294]. Even though the content aspect of construct validity can be determined by the professional judgement of experts [295], it is still possible to assess it using the Rasch model. Fit indices can measure the relevance of the test content to the domain under investigation; items that do not fit may be irrelevant to that domain. On the other hand, a person-item map and item strata are very useful in determining the representativeness of the items [295]. The test must include two levels of difficulty to be considered a suitable representation of the intended domain [289]. However, large gaps in the item difficulty hierarchy indicate that an area of the domain has not been covered [296]. Moreover, the technical quality of the test can be checked using item-measure correlations, as these can highlight poor-quality or mistyped items [294].

Person fit statistics and multiple-choice distractor analysis are crucial measures for the substantive aspect of construct validity. Person fit statistics measure the extent to which a person's response pattern matches the pattern predicted by the model [289]. Guessing or carelessness in responses indicates a misfit in the analysis. Multiple-choice distractor analysis in the Rasch model includes p -values, choice means, and distractor-measure correlations. The p -value measures how often each distractor has been chosen. Since the ideal situation where each distractor is expected to be equally attractive is impossible, p -values are used to detect faulty distractors [294]. Choice means refers to the average number of respondents who choose each distractor; it shows its discrimination power. Ideally, distractors should differentiate between high and low-ability respondents.

Item-measure correlations can also be used to prove the unidimensionality of the test [294], which represents the structural aspect of construct validity. In other words, fit statistics can indicate that single scoring is not appropriate, and separate scores should be defined for each dimension. The generalisability aspect of construct validity can be examined by the invariance of item measures across different populations or over time [295]. It is also possible to use the invariance of person measures across different sets of test items. The ability of the test to detect within-individual or between-group changes can determine the external aspect of construct validity [287]. Thus,

the person-item map and person-strata index are very useful for deciding on this aspect [297].

The only aspect of construct validity that the Rasch model does not provide a straightforward calculation for is the consequential aspect. However, checking item bias and investigating differential test functioning or even a closer look at the person-item map can help to ensure the consequential aspect of construct validity [295].

The Rasch model approach has been considered a reliable and effective framework for verifying the construct validity of multiple-choice tests [295, 298, 291]. It has been used in many research areas, including education, medicine, and others [299]. For example, many studies in the education field have applied the Rasch model to assess the validity of measurement and testing instruments in language teaching [300]. Specifically, a number of researchers have used the Rasch model to validate vocabulary tests, such as Baghaei and Amrahi [295], Beglar [301], Runnels [302], Maskor et al. [303], among others.

4.3.3 Validation Methods and Tools

The data used to validate the test was collected as part of the research case studies in the first year of the data collection process. CTT results for 150 students from Case Study One (China) and Case Study Two (Saudi Arabia) were analysed using Winsteps software [304]. Several statistical methods were applied to the data based on the Rasch model, including item and person fit measures, item and person strata and others.

4.3.4 Validation Results

In Rasch model analysis using Winsteps, unexpected response patterns to the test can be detected by INFIT statistics, while OUTFIT can detect guessing or mistakes [305]. Fit statistics consist of mean-square values (MNSQ) and z-standardized scores (ZSTD). An MNSQ greater than 1.0 indicates underfit to the model, meaning that the data are less predictable than expected. On the other hand, an MNSQ less than 1.0 indicates overfit to the Rasch model, meaning that the data are too predictable. In other words, items with an MNSQ less than 1.0 do not contradict what is already known but do not add anything about what needs to be known, making them inefficient. This is why overfit is less important than underfit in test validation measures [305].

Moreover, Winsteps software tests the hypothesis that $MNSQ = 1.0$, meaning that the randomness in the data is equal to the randomness predicted by the Rasch model.

The result of this hypothesis testing is presented as ZSTD values. In general, ZSTD values greater than ± 1.96 indicate that the null hypothesis of 'model fit' is rejected. For short multiple-choice tests, acceptable values are between 0.7 and 1.3 for MNSQs and between -2.0 and 2.0 for ZSTDs, for both INFIT and OUTFIT statistics [306].

The validation process for the CTT began with item fit statistics; these were analysed to check the content and structural aspects of validity. Table 4.4 provides a summary of item fit statistics for all test measures. The item strata separation shown in the table can be used to check the content aspect of validity, particularly representativeness. A test should have at least two levels of item difficulty to be considered representative [289]. Clearly, this test has a good level of representativeness, with a separation value of 4.03. Item reliability stands at 0.94, indicating that the sample size was large enough to ensure the item difficulty hierarchy [305]. It is also noticeable that the test has acceptable mean values of INFIT and OUTFIT for MNSQ and ZSTD measures.

Item	Total	Count	INFIT		OUTFIT	
			IMNSQ	ZSTD	OMNSQ	ZSTD
Mean	94.5	149.9	1.00	.1	.98	.0
P.SD	24.0	.2	.11	1.5	.20	1.6
# Input	30	Separation	4.08		Reliability	.94

Table 4.4: Summary of Item Statistics

Figure 4.3 represents the person-item map of 150 responses to the CTT using Winsteps software [304]. The most unique feature of this map is the placement of persons and items on the same scale, which is very useful for comparisons [307]. In other words, the persons' ability and the items' difficulty are mapped using unified logit measures derived from the raw scores in the analysed data. The measures column at the far left shows the logit units that represent the levels of the students' ability and the items' difficulty [305]. The right-hand side of the map shows the test items (Q) arranged by difficulty level, from hardest at the top to easiest at the bottom. Similarly, the left-hand side of the map shows the positions of the test-takers, indicated by (X), based on their ability level. Students who demonstrate a higher ability level appear at the top of the figure, while those who demonstrate a lower ability level appear at the bottom. Similarly, the items with high difficulty levels appear at the top, and the easier items appear at the bottom of the map.

In Figure 4.3, the test items show an acceptable distribution across the different levels of students' abilities. However, Q13 and Q8 are above most students' ability levels, while Q1 and Q3 are too easy for the tested sample. It is also shown on the left side of the

map that some students are above the test level; these students are the ones who scored 90% or above. The only noticeable gap, although not huge, in the map is between Q26 and Q8, indicating that there are no corresponding questions for students at this ability level. To address this gap, two medium-to-high-difficulty questions could be added. Overall, the test can be considered to have an acceptable level of representativeness.

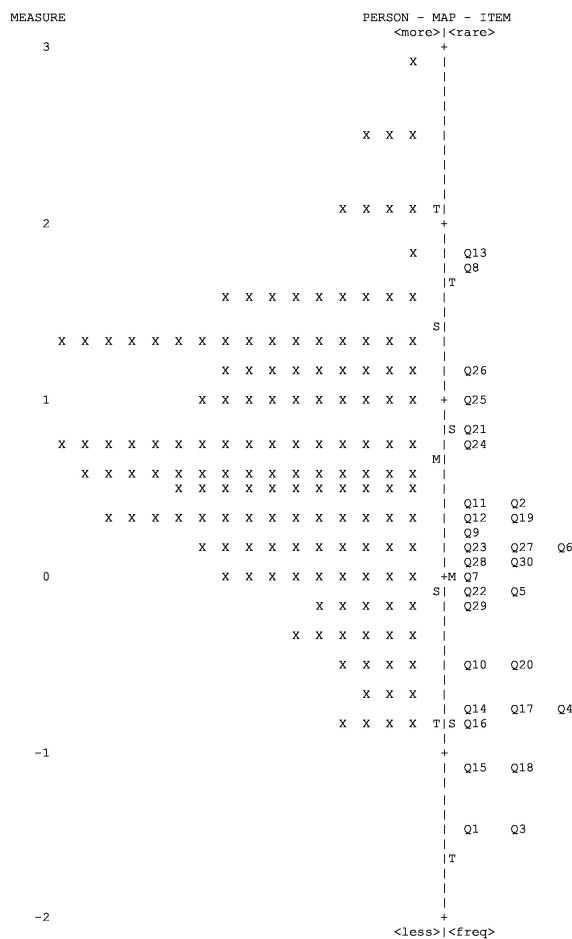


Figure 4.3: Person-Item Map

The detailed output of item fit statistics has been reviewed to confirm the relevance of the items to the intended construct. The results indicate that most test items fall within the acceptable range of MNSQ and ZSTD values. The full item fit statistics output for the 30 CTT items is too extensive to be included here but is available in Appendix E. Instead, Table 4.5 presents the items that fall outside the acceptable range of MNSQ and ZSTD values. Items Q8, Q26, Q21, and Q29 have OUTFIT z-standardized scores of 2.41, 2.17, 2.13, and -2.14, respectively, which are slightly beyond the acceptable range.

According to Linacre [305], more attention should be given to INFIT indices rather than OUTFIT, as they identify misfitting items in the regions that should be properly covered by the test. Thus, items Q8, Q26, Q21, and Q29 can be considered a good fit for the test, as they exhibit acceptable values in all other statistical measures. The exceptions are questions Q13, Q11, and Q6, which represent irrelevant or misfitting items. Overall, 27 out of 30 items have fair fit statistic values, indicating an acceptable level of relevance.

Item-measure correlations provide evidence for technical quality (part of the content aspect of validity) and unidimensionality, which represents the structural aspect [308]. They indicate the direction of what the item is measuring. Item correlations close to zero suggest that the item is either very easy, very difficult, or measuring the construct in a different way from other items [294]. On the other hand, negative correlations indicate that an item opposes the direction of construct measurement. The results of item statistics for all test items can be found in Appendix E; however, items with unacceptable values are listed in Table 4.5. All other items have been thoroughly reviewed, confirming that their measurements fall within the acceptable range. Once again, Q13 shows a negative correlation (CORR.), indicating a possible technical issue (e.g. ambiguity). Additionally, Q8 and Q11 have correlation values very close to zero. All other items exhibit an acceptable level of correlation, supporting the technical and structural aspects of validity.

ITEM	INFIT		OUTFIT		PTMEASUR-AL	
	IMNSQ	ZSTD	OMNSQ	ZSTD	CORR.	EXP.
Q6	.85	-2.63	.83	-2.28	.51	.31
Q8	1.19	2.08	1.31	2.41	.03	.31
Q11	1.20	3.55	1.23	3.22	.06	.32
Q13	1.29	2.81	1.64	4.21	-.15	.30
Q21	1.11	1.96	1.14	2.13	.18	.33
Q26	1.10	1.53	1.18	2.17	.17	.32
Q29	.86	-1.96	.79	-2.14	.49	.30

Table 4.5: Selected Item Statistics (The full result is available in Appendix E)

To sum up, the content aspect of validity, including relevance, representativeness, and technical quality, has been checked using different types of statistics. Item strata separation and the person-item map provided evidence that the test has a good level of representativeness. Moreover, the detailed output of item fit statistics, particularly MNSQ and ZSTD values of INFIT and OUTFIT, has been examined to ensure item relevance. Based on the results, 27 out of 30 questions show an acceptable level of relevance to the construct domain. Finally, item-measure correlation values for all items have been

checked to evaluate technical quality. The results indicate that three questions might have some type of technical issue, while the remaining 27 questions are satisfactory. Therefore, the test can be considered to have an acceptable degree of relevance, representativeness, and technical quality, supporting the content aspect of validity.

A distractor analysis has been conducted to examine the substantive aspect of validity. This analysis shows the mean of the ability estimates of all test-takers who have chosen the correct and wrong options. It is expected that the ability mean of the correct answer should be greater than the wrong answers, otherwise, the distractor is not functioning in the intended way. The output of this analysis is too lengthy to be included here; however, it is available in Appendix F. In this analysis, Q13's ability mean values are 0.48 and 0.73 for the correct and incorrect answers, respectively. This indicates that the incorrect answer options for Q13 are not functioning correctly. The results have been carefully examined, and all other items show acceptable values. Thus, the substantive aspect of validity for this test has been confirmed.

This test has been completed by two groups of students, who are considered to be from different populations due to their native languages. The invariance of item measures across groups has been used to examine the generalisability aspect of validity. The differential test functioning (DTF) plot in Figure 4.4 shows that all item measures fall within the control lines. Clearly, the test is valid regarding the generalisability aspect. Moreover, no significant DTF has been found between groups (Figure 4.4), indicating that there is no evidence of biased items. Thus, the consequential aspect of validity is ensured, and any decision made based on this test result is expected to be fair.

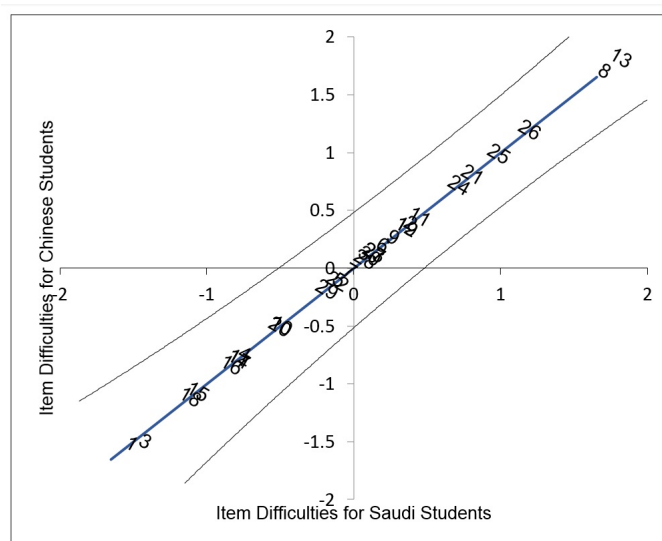


Figure 4.4: Between-groups differential test functioning (DTF)

To ensure the external aspect of validity, the person separation strata should be close to one [291]. For this test, the person separation stands at 1.30 (Table 4.6), which is considered an acceptable value. Moreover, a test that can differentiate between students with low and high levels of ability can be considered as having external validity [295]. On the person-item map (Figure 4.3), students with very high ability levels appear at the top, while students with low ability levels appear at the bottom, representing a hierarchy in ability levels. Therefore, it can be said that this test has an acceptable level of differentiation between various levels of student ability.

Item	Total	Count	INFIT		OUTFIT	
			IMNSQ	ZSTD	OMNSQ	ZSTD
Mean	18.9	30.0	.99	.1	.98	.0
P.SD	4.1	.2	.19	1.2	.29	1.2
# Input	150	Separation	1.30	Reliability		.63

Table 4.6: Summary of Person Statistics

4.3.5 Validation Summary and Discussion

Overall, the computer terminology test demonstrates an acceptable level of validity in this analysis. Regarding the content aspect, 90% of the test items are relevant to the intended construct based on fit statistics, with exceptions being Q13, Q11, and Q6. It can also be said that the test has a fair level of representativeness based on item separation statistics and the person-item map.

Additionally, 29 out of 30 test items exhibit acceptable levels of technical quality and structural validity based on an examination of item-measure correlations, with Q13 being the exception. The same 29 test items also demonstrate an acceptable level of substantiveness based on multiple-choice distractor analysis, again with the exception of Q13. Furthermore, the invariance of item measures across two different populations confirms the generalisability aspect. Similarly, the differential test functioning shows no evidence of bias, supporting the consequential aspect.

However, a closer examination of certain test items is necessary, namely Q13, Q11, and Q6. The most concerning case is Q13; its INFIT and OUTFIT measures are high, its item-measure correlation is negative, and its ability mean falls outside the acceptable range. It also appears at the very top of the person-item map, indicating a high level of difficulty. In the future, it may be advisable to consider removing this question or replacing it with a new one.

On the other hand, Q11 and Q6 have only one issue: high ZSTD values in the INFIT and OUTFIT measures. This indicates that these questions could be measuring something other than the intended content and might need to be revised [295]. Additionally, Q8 has a high ZSTD OUTFIT value, a near-zero item-measure correlation, and a high level of difficulty according to the person-item map. However, as noted by Linacre [305], a misfit OUTFIT value is not a significant concern due to the test's sensitivity to guessing and carelessness. The low correlation measure could be a result of the high level of difficulty. Given this, Q8 can be considered the most difficult question on the test. Nonetheless, this question can still be a good fit if the test sample represents a higher level of ability; therefore, there is no need to remove this item from the test.

Recommendations to enhance the test can be made based on the person-item map. Since Q1 and Q3 tend to be very easy, they could be revised or replaced to increase the test's level of difficulty. However, this is not recommended if the test is to be administered to students with no basic knowledge of computer science. Additionally, more items with a medium-to-high level of difficulty could be added to cover a wider range of ability levels. However, the test's duration should be considered when adding more questions. Moreover, the timing of the test is important to consider. For example, administering this test in the first few weeks of a CS1 course will make it more difficult compared to administering it in the middle or at the end of the course. The final recommendation concerns Q13, which is the only question with major issues. Omitting this question could positively impact the test's validity. It is also possible to carefully review it or replace it with a different question.

The level of validity is highly acceptable, and any changes will not guarantee a higher or the same level of validity. Additionally, the validation results presented in this chapter will not apply to the test if any question is modified, removed, or added, as another validity check will be required. Therefore, it was decided that the test would be used as it is in the second data collection period, with no modifications, to ensure the consistency of the data. Finally, although a large sample of participants from different backgrounds and institutions has been used in the validation of this test, the results are still limited to NNES students and the online teaching and testing method.

4.4 Conclusion

This chapter outlined the process of compiling the CS1 word list, which forms the foundation for building the CTT. Following this, the process of designing and developing the computer terminology test for CS1 courses was explained. This chapter also

discussed in detail the methods, tools, and results related to assessing the different aspects of the CTT's validity. Additionally, recommendations for enhancing the test quality based on the validation results were provided for future research. The CS1 word list and terminology test produced can benefit CS1 teachers, course designers, students, and researchers interested in this topic. The full CS1 word list is included at the end of section 4.2 in this chapter, and the CTT can be found in Appendix B at the end of this thesis.

After ensuring the validity of the CTT, it was used to measure the students' knowledge of computer terminology. The results were then used to determine the impact of this factor on their academic performance in the CS1 course. The correlation results are presented in the next chapter, along with the other two factors studied. Moreover, the same test was used to investigate the efficacy of the online managed and peer-reviewed terminology translation activity in improving students' knowledge of such terminology. The design and evaluation of this learning activity are presented in chapter 6 of this thesis.

DATA ANALYSIS AND RESULTS

5.1 Introduction

This chapter presents the data analysis procedure and results. The results of each case study are presented separately, followed by a cross-case analysis. The data collected using the background survey was used to provide an overview of the case study as well as explore the participants' perspectives on the topic. Correlation and multiple linear regression results are also presented in this chapter. The outcome of the analysis was used to test the research hypotheses at the end of this chapter¹.

5.2 Data Analysis Procedure

By this stage, the data had already been cleaned, coded, de-identified, and checked in preparation for the analysis process, as explained in section: 3.4.2 of chapter 3. The analysis results begin with a general background on the participants in each case study. The data used in this background analysis was collected through the background survey and includes demographic information and academic background. The aim is to provide an overview of the selected case study used in this research.

The survey also included questions aimed at exploring the students' perspectives on using English in class and learning English-based programming languages. Under-

¹The related published work to this chapter includes:

- Alaofi, S., and Russell, S. **'The Influence of Foreign Language Classroom Anxiety on Academic Performance in English-based CS1 Courses.'** in *Proceedings of the United Kingdom & Ireland Computing Education Research Conference (UKICER)*, article, 2022, 7 pages. [Online].
- Alaofi, S., and Russell, S. **'The Use of English Language to Teach CS1 to Non-Native English Speakers: Students Perspective.'** in *Proceedings of the ACM Global Computing Education Conference (CompEd)*, article, 2023, 7 pages. [Online].

standing the students' perspectives can help in interpreting the correlation results. The students' perspectives based on the survey are presented after the background details.

In preparation for the correlation analysis, a scatterplot matrix for all six dependent and independent variables was generated. The purpose of the scatterplots was to explore the existence of a linear relationship between the variables in each case study. Following this, a correlation analysis was conducted to assess the statistical significance of the relationships observed in the scatterplots.

It is possible that not all students participated in every questionnaire and test used in this study. Therefore, pairwise deletion or available-case analysis was applied during the correlation analysis to minimise data loss. This explains the differences between the total number of participants in the research instruments and the number that appeared in the correlation analysis results.

Multiple linear regression was also conducted following the correlation analysis to examine the ability of the independent variables to predict NNES students' academic performance in the CS1 course. The regression analysis results are explained in detail for each case study.

Following the analysis results of each case study separately, a cross-case analysis is presented. This type of analysis provides a detailed comparative examination, including the case study background, students' perspectives, correlation results, and regression analysis outcomes. The aim of the comparative analysis is to explore the similarities and differences between the two case studies, which can help in understanding common patterns observed in the results. Finally, the research hypotheses presented in chapter 1: section 1.5 are tested based on the analysis outcomes.

5.3 Case Study One (China)

5.3.1 The Participants

The number of responses collected using each research instrument from students at Beijing-Dublin International College (BDIC) in China is shown in Table 5.1. All participants enrolled in the selected CS1 courses were undergraduate students in the School of Computer Science or the School of Engineering.

Based on the background survey completed by 276 students, 69 identified themselves as female, 204 as male, and 3 students chose not to specify a gender. Additionally, 271 students were aged between 18 and 23, while five students preferred not to disclose

their ages. According to their responses regarding native language, all participants were native Chinese speakers.

Only 29 students had attended a high school that used EMI, and 11 had completed an English course in an English-speaking country. In the academic background section, 269 students had completed a mathematics subject in high school, 259 had studied science subjects (Chemistry, Physics, or both), 267 had taken English as a foreign language, and 122 had completed a CS basics subject. Regarding programming experience, 41 students had completed formal programming courses before this course, and 46 had learned some programming independently.

Academic Year	Background Survey	SET	CTT	FLCAS
20/ 21	144	110	92	116
21/22	132	80	92	123
Total	276	190	184	239

Table 5.1: Number of participants - Case Study One (China)

The participants' level of programming experience prior to the CS1 course, based on the self-reported scale, is as follows: 126 had no experience, 86 were novices, 47 were at an intermediate level, 16 were advanced, and one student was at the expert level. The scale descriptions provided in this question are as follows:

- **No experience** (*Never learned or used any programming languages.*)
- **Novice** (*I have some knowledge of basic programming concepts, but I am not able to write or execute any programming code without help.*)
- **Intermediate** (*I am able to write and execute a simple programming code by myself but not able to correct the errors appear in my code without help.*)
- **Advanced** (*I am able to write and execute more complicated programming code and correct most of the errors appear in my code.*)
- **Expert** (*I am able to develop a complete piece of software or application using a programming language and correct all errors appear in my code.*)

Finally, their average English proficiency level is 54.65 (SD 11.69) based on the standard English test completed by 190 students. This places the students' English proficiency level in this case study between intermediate (B1) and advanced (C1) levels according to the EF scale system [239].

5.3.2 The Students' Perspectives

5.3.2.1 The difficulties students face when using English in the CS1 class

In this question, the students were asked to choose how often they faced some of the difficulties related to using English in the CS1 course. For this question, a list of eight difficulties that appeared in the literature was selected. Each item in this question was provided with five answer options, including '*never*', '*rarely*', '*sometimes*', '*often*' or '*always*'. The results in Figure 5.1 show that the top three difficulties faced by approximately 28% of the students, often or always, include asking or answering questions in English, understanding programming concepts when explained in English, and understanding some English computer terminology used in the class. Giving the wrong answer due to misunderstanding the questions was faced by 24% of the students often or always.

On the other hand, tasks such as understanding or following up with the teachers when they speak in English, reading or understanding exams or assignments, and reading or understanding lecture notes and learning materials were less difficult. For these tasks, only 16–18% of the students said they faced difficulty with them often or always. Regarding the difficulty in writing or explaining answers in English, responses were more neutral, with 43% selecting the option sometimes, 38% choosing never or rarely, and 19% selecting often or always. In general, between 35–45% of the students said they faced difficulties sometimes in all of the previously mentioned tasks.

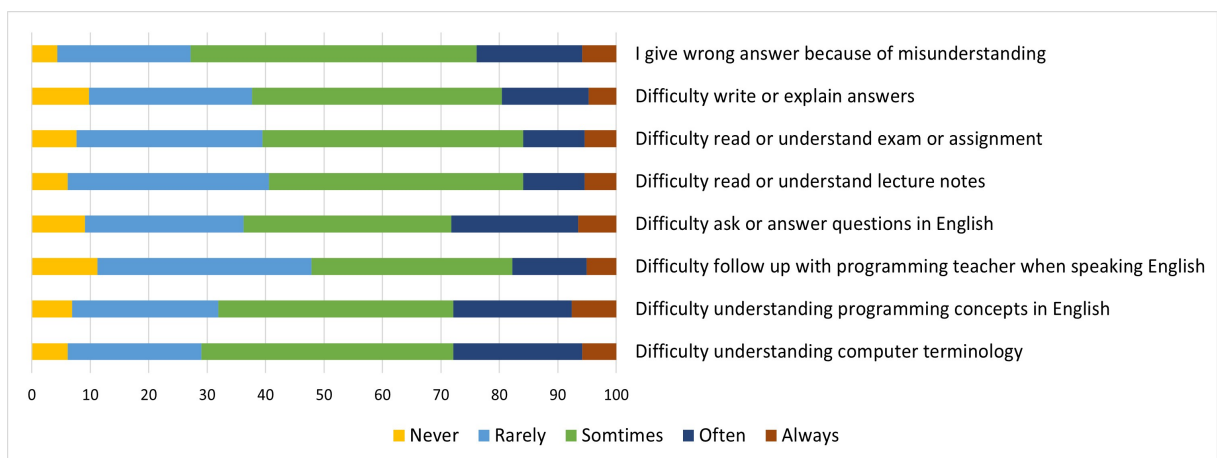


Figure 5.1: Difficulties with using English in CS1 class - Case Study One (China)

The follow-up open-ended question gave the students the opportunity to add or explain any difficulties they were facing regarding using English in CS1 class. 30 students responded to this question, which represents about 11% of the total responses to the survey. However, some responses mentioned more than one difficulty, so they

contributed to multiple themes at the same time. This explains why the total number of responses across all themes is greater than the number of students who answered the question.

Table 5.2 shows the main themes that emerged from the collected responses. 50% of the responses represent difficulties related to general English comprehension. More than a quarter of the collected responses (27%) stated problems in understanding English terminology with computer science meanings and other academic vocabulary. 23% of the students who responded to this question said that they face difficulties in understanding programming concepts when explained in English. Another 20% of the responses expressed difficulties in finding or understanding the translations of technical terms in their native language. 13% of the students who answered this question said they face difficulties in understanding learning materials, assignments, and exams provided in English. Lastly, 10% mentioned difficulties in learning programming through online classes, which was the only available option at the time of data collection due to the COVID-19 pandemic.

Themes	<i>n</i> of responses contributing to the theme (%)
General English language difficulty	15 (50%)
Difficulty understanding academic/ professional/ CS vocabulary	8 (27%)
Difficulty understanding programming concepts in English	7 (23%)
Difficulty finding or understanding translations/ meanings in native language	6 (20%)
Difficulty understanding learning materials/ assignments/ exams	4 (13%)
Online learning difficulty	3 (10%)

Table 5.2: Difficulties with using English in CS1 class - Case Study One (China)

5.3.2.2 The difficulties students face when using English-based programming languages.

Another question in the survey asked the students to select the difficulty level of tasks related to learning and using English-based programming languages. The question includes nine tasks or activities, and the options range from '*very easy*' to '*very difficult*', with a neutral option in the middle. The results in Figure 5.2 show the top four diffi-

cult tasks when learning English-based programming languages, which 28–30% of the students find difficult or very difficult. These tasks include reading error messages, understanding and using error messages, using the English interface of the programming software used for coding (IDE), and understanding or using the IDE hints, tips, and help commands in the English interface. Additionally, approximately 23% of the students find the following two tasks either difficult or very difficult: intuitively understanding the programming language keywords’ meaning and functionality, and understanding the relationship between the programming language keywords and their meaning or functionality.

On the other hand, only about 16% of the students find reading and writing code using English letters/alphabet difficult or very difficult. The least difficult task was remembering the correct spelling of the programming language keywords, with only 13% finding it difficult or very difficult and about 43% finding it easy or very easy.

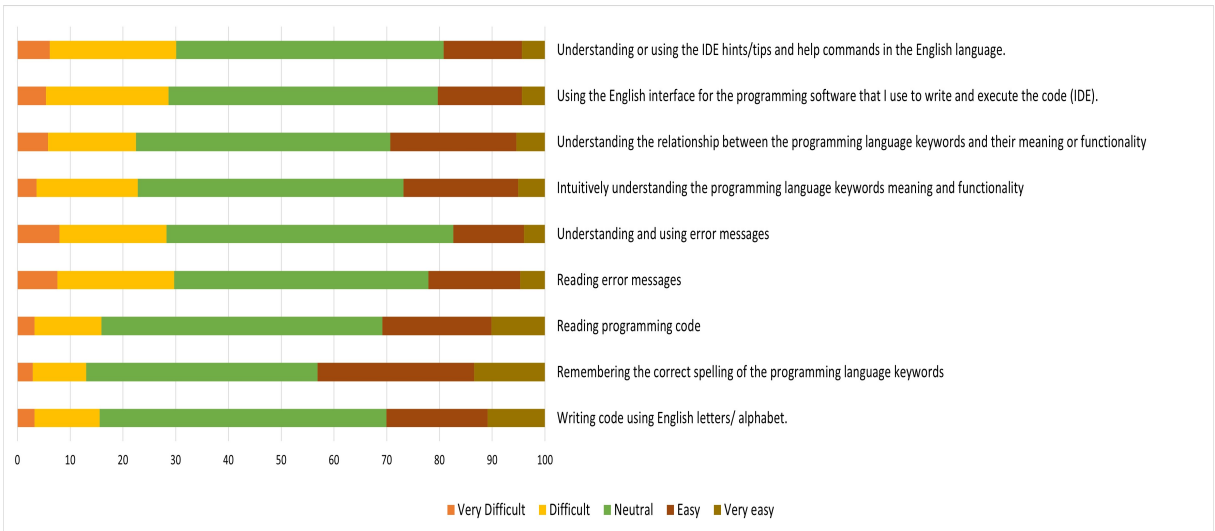


Figure 5.2: Difficulties with using English-based programming languages - Case Study One (China)

After the multiple-choice question, the students were given the option to add or explain the difficulties they face when learning English-based programming languages in their own words. Table 5.3 shows the main themes of the responses given by 12 students. The top three difficulties, which represent 25% of the total responses, were reading and comprehending programming code, using and understanding error messages, and problem-solving and computational thinking difficulties. The last three difficulties, mentioned in 8% of the responses, include difficulty remembering or distinguishing programming keywords, following the programming language syntax, and online learning difficulties.

Themes	<i>n</i> of responses contributing to the theme (%)
Difficulty reading/ comprehending/ expecting the outputs of programming code	3 (25%)
Difficulty using/ understanding error messages	3 (25%)
Difficulty in problem-solving/ computational thinking/ logical thinking	3 (25%)
Difficulty remembering/ using/ distinguishing programming keywords	1 (8%)
Difficulty following the programming language syntax/ format	1 (8%)
Online learning difficulty	1 (8%)

Table 5.3: Difficulties with English-based programming languages - Case Study One (China)

5.3.2.3 To what extent do students consider the use of the English language as a barrier to learning programming?

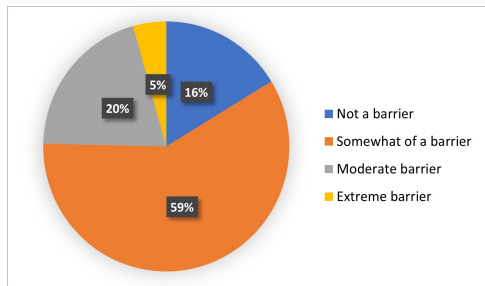
The result in Figure 5.3a shows that 56% of the surveyed students find the use of English in the CS1 course somewhat of a barrier to learning programming. 20% believe it represents a moderate barrier, while only 5% see it as an extreme barrier. For 16% of the students, using English in CS1 class is not a barrier at all.

5.3.2.4 What is the impact of using English on the level of the CS1 course difficulty as perceived by students?

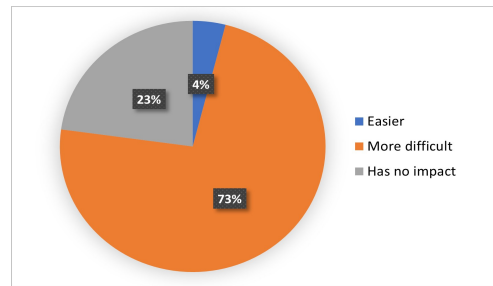
The result in Figure 5.3b shows that the majority of the surveyed students (73%) believe that the use of English in the CS1 course makes it more difficult, while only 4% think it makes the course easier. For 23% of the students, the use of English in the CS1 course has no impact on the difficulty level of the course.

5.3.2.5 What language(s) do students prefer to use to study programming?

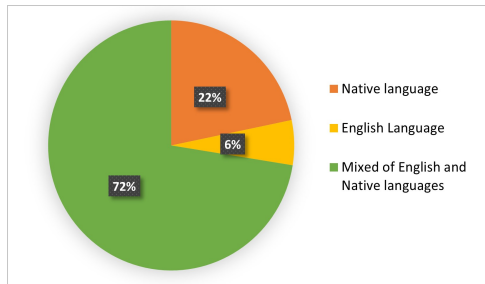
A large number of students (72%) prefer to use a mix of English and their native language in CS1 classes, as shown in Figure 5.3c. Only 6% would like to continue using only English, while 22% express their desire to use only their native language in the CS1 course.



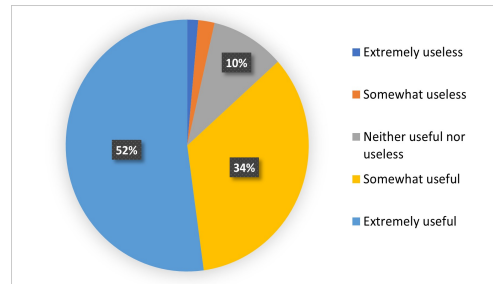
(a) Is English a barrier for learning



(b) Impact of English on course difficulty



(c) Preferred language(s) to use in the course



(d) Perceived usefulness of using English

Figure 5.3: Students' perspectives about using English in CS1 course - Case Study One (China)

5.3.2.6 Do students consider the use of English in this course as beneficial for their future study and career?

In the end, more than half of the students (52%) believe that using English in the CS1 course is extremely beneficial for their future studies or career (Figure 5.3d). 34% think it is somewhat useful, while 10% have a neutral opinion, seeing it as neither useful nor useless. Only 1% to 2% of the students believe that using English in the CS1 course is not useful at all for their future, as they selected one of the last two options: '*somewhat useless*' or '*extremely useless*' to answer this question.

5.3.3 Correlation Analysis Results

The scatterplot matrix in Figure 5.4 explores the possible correlations between the dependent and independent variables. The first row (A) shows a possible positive association between the Standard English Test (SET) scores and the students' academic performance in the CS1 course. These associations seem weak; however, the association between SET and the overall CS1 score appears stronger than the other two academic performance indicators.

The second row (B) shows a possible positive association between the students' performance in the Computer Terminology Test (CTT) and their academic performance in CS1 courses. The relationship between CTT and the overall CS1 score appears stronger than that with the other two academic performance indicators.

The third row (C) explores the possible association between the Foreign Language Classroom Anxiety Scale (FLCAS) and academic performance in the CS1 course. It seems that there is no correlation between this factor and students' academic performance in the final exams. However, this factor could have a possible negative relationship with academic performance in assignments and overall score in the CS1 courses.

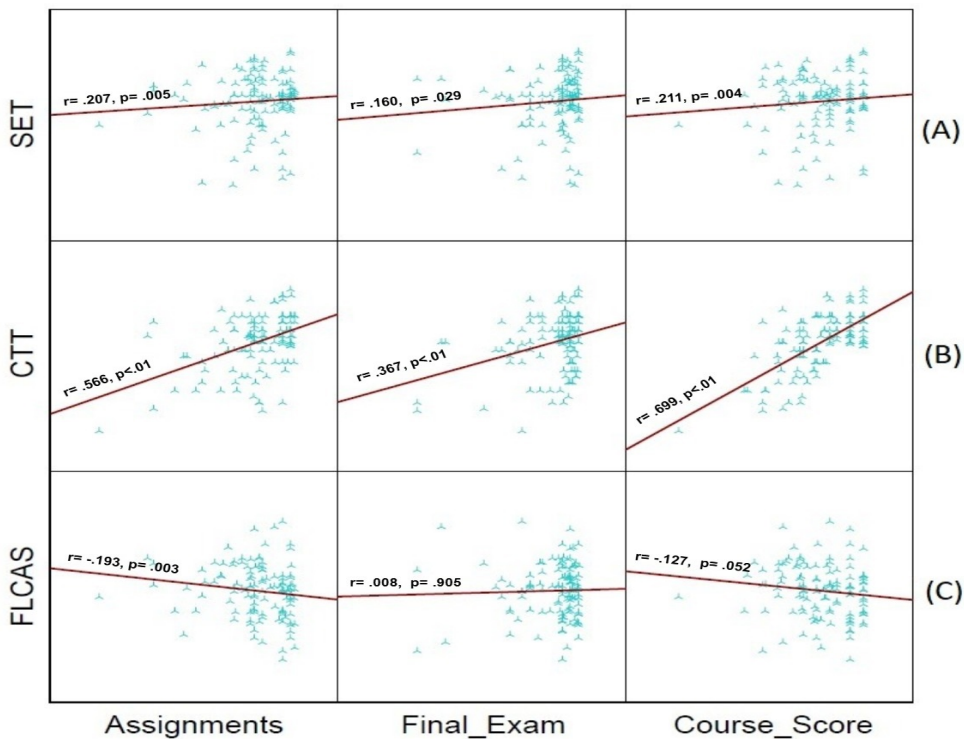


Figure 5.4: Scatterplots matrix - Case Study One (China)

The next step is to test the strength of the correlations that appeared in the scatterplots above using Pearson's coefficient correlation (r) analysis. The following Table 5.4 shows the analysis results, including r , p -value, and the number of cases used in each correlation analysis. The strength of the correlation has been described as weak ($-.20 \leq r \leq .20$), moderate ($.20 < r \leq .50$), and High ($r > .50$) [309]. In the correlation table, highlighting colours were used to show the strength of the correlation. The lighter the colour, the weaker the correlation is considered.

The result shows a low, significant positive correlation between English proficiency level, as measured by the SET, and all three assessment aspects for the CS1 course. The students' performance in the SET seems to be equally correlated with both academic

performance in assignments and overall CS1 course score ($r=.207$, $r=.211$, $p<0.01$, respectively). A lower correlation exists between the SET and final exam score ($r=.160$, $p<0.05$).

The correlation results for the CTT and all three aspects of the CS1 assessment show a moderate to high positive correlation. The assignments score shows a higher correlation with CTT ($r=.566$, $p<0.01$) than the final exam score ($r=.367$, $p<0.01$). However, the highest correlation was with the overall CS1 course score ($r=.699$, $p<0.01$).

The above positive correlations suggest that the students' academic performance in the CS1 course increases in parallel with increased performance in the SET and CTT.

Finally, the FLCAS shows a low but significant negative correlation with CS1 assignment score ($r=-.193$, $p<0.01$). There is no correlation between FLCAS levels and final exam score or the overall CS1 course score. The negative correlation indicates that the students' academic performance in the CS1 course, especially in assignments, decreased as their scores in FLCAS increased.

	Assignments Score			Final Exam Score			Overall Course Score		
	r	p	n	r	p	n	r	p	n
SET	0.207**	0.005	187	0.160*	0.029	187	0.211**	0.004	186
CTT	0.566**	0.001	184	0.367**	0.001	184	0.699**	0.001	183
FLCAS	-0.193**	0.003	237	0.008	0.905	238	-0.127	0.052	237

** Correlation is significant at the 0.01 level.
* Correlation is significant at the 0.05 level.
Correlation Strength: Weak ($-.20 \leq r \leq .20$) moderate ($.20 < r \leq .50$) High ($r > .50$)

Table 5.4: Correlation Results - Case Study One (China)

5.3.4 Multiple Linear Regression Analysis Results

The aim of multiple linear regression is to find the best predictor for the students' academic performance in the CS1 course from the three independent variables under investigation. The summary of the linear regression model in Table 5.5 shows that 56.6% ($r^2=.566$) of the variance in the students' academic performance in the CS1 course can be explained by the independent variables (i.e. SET, CTT, and FLCAS). This result is satisfactory to proceed with the linear regression analysis results.

Country	Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
CN	1	0.752 ^a	0.566	0.555	9.64

a. Predictors: (Constant), SET, FLCAS, CTT
b. Dependent Variable: Course_Score

Table 5.5: Model Summary^b - Case Study One (China)

The ANOVA analysis results in Table 5.6 show that the prediction model can be reliably used to predict the students' academic performance in the CS1 course using the three independent variables (Sig. < 0.01). However, the ANOVA table does not explain how each independent variable individually predicts the students' academic performance.

Country	Model		Sum of Squares	df	Mean Square	F	Sig.
CN	1	Regression	14670.58	3	4890.194	52.619	.000 ^b
		Residual	11245.3	121	92.936		
		Total	25915.88	124			

a. Dependent Variable: Course_score
b. Predictors: (Constant), SET, FLCAS, CTT

Table 5.6: ANOVA^a - Case Study One (China)

The ability of each individual independent variable to predict the students' academic performance in the CS1 course is shown in Table 5.7. The most important value in this table is the *p*-value (Sig.) in the last column. This value should be < 0.05 to reject the null hypothesis and accept the independent variable as a reliable predictor for the students' academic performance. In the same table, only CTT shows a *p*-value < 0.05, while the other two independent variables show *p*-values > 0.05. The B value under the Unstandardized Coefficients column in the same table is 0.656 for CTT. The explanation is that for every point increase in the CTT score, a 0.656 unit increase in the overall CS1 course score is predicted.

Country		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
CN	(Constant)	44.4	6.36		6.982	.000
	CTT	0.656	0.055	0.735	11.885	.000
	FLCAS	-0.054	0.036	-0.092	-1.517	0.132
	SET	0.015	0.076	0.012	0.191	0.848

a. Dependent Variable: Course_Score

Table 5.7: Multiple linear regression coefficients^a - Case Study One (China)

To summarise the multiple linear regression results, it is possible to say that CTT is the only independent variable that can predict the students' academic performance in the CS1 course. Neither SET nor FLCAS has the ability to reliably predict the students' academic performance in CS1 courses, as they show no significant values. Consequently, running simple linear regression with CTT as an independent variable and

overall course score as a dependent variable could be more efficient. In the result of the simple linear regression, $r^2 = .488$, which means that 49% of the variability in the total CS1 score can be predicted from the relationship with the CTT. Also, the B value is equal to .708, which means that for every point increase in the CTT score, a 0.7 unit increase in the overall CS1 course score can be predicted. This result is very close to the result obtained from the multiple linear regression.

5.4 Case Study Two (Saudi Arabia)

5.4.1 The Participants

The number of responses collected using each research instrument from students at Shaqra University in Saudi Arabia is shown in Table 5.8. All participants enrolled in the selected CS1 courses were undergraduate students in the Department of Computer Science.

Based on the background survey completed by 117 students, 37 identified themselves as female, and 80 as male. Also, 116 students aged between 18 and 23, while one student preferred not to answer. Additionally, 116 students were aged between 18 and 23, while one student preferred not to disclose their age. According to their responses regarding native language, all participants were native Arabic speakers.

Only two students had attended a high school that used EMI, and five students had completed an English course in an English-speaking country. In the academic background section, 114 students had completed a mathematics subject in high school, 114 had studied science subjects (Chemistry, Physics, or both), 117 had taken English as a foreign language, and 114 had completed a CS basics subject.

Regarding programming background, 99 students stated that the current CS1 course was their first formal programming course, while 13 had learned some programming independently before this course. Based on the self-reported programming experience scale, 59 students had no prior experience, 42 identified as novices, 15 as intermediate, and one as advanced. The scale description provided with this question is presented in Section 5.3.1.

Finally, the average English proficiency level for this group was 38.21 (SD 11.89) based on the standard English test completed by 113 students. This places the students' English proficiency level between Beginner (A1) and Intermediate (B1) levels according to the EF scale system [239].

Academic year	Background Survey	SET	CTT	FLCAS
20/21	61	49	59	47
21/22	56	64	58	67
Total	117	113	117	114

Table 5.8: Number of participants - Case Study Two (Saudi Arabia)

5.4.2 The Students' Perspectives

5.4.2.1 The difficulties students face when using English in the CS1 class

As explained in Case Study One (China), this question asked students to choose how often they face some of the difficulties related to using English in the CS1 course. Each of the eight items in this question had five answer options ('never', 'rarely', 'sometimes', 'often', 'always'). Figure 5.5 shows that the top four difficulties students face when using English in CS1 class include writing or explaining answers in English, asking or answering questions in English, following up with the teacher when speaking English, and understanding programming concepts explained in English. Between 28% to 30% of the students face these four difficulties often or always. The difficulty in understanding programming concepts explained in English affects approximately 50% of the students sometimes, whereas this number is 30% for the other top three difficulties.

Some other difficulties, such as reading or understanding exams or assignments, reading or understanding lecture notes, and understanding computer terminology, are less frequent, with only about 17% of students facing them often or always. However, 50% of the students reported sometimes having difficulty understanding computer terminology. Only about 12% said they often or always give a wrong answer due to misunderstanding questions asked in English, while about 55% said they never or rarely face this difficulty.

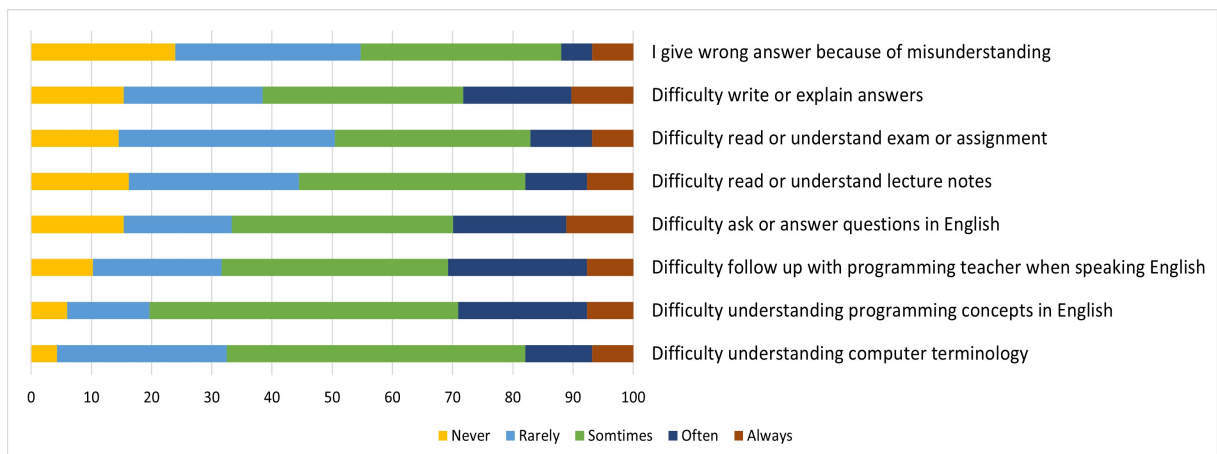


Figure 5.5: Difficulties with using English in CS1 class - Case Study Two (Saudi Arabia)

The open-ended question following the multiple-choice questions aimed to collect qualitative data to better understand the students' perspectives. In this case study, only six students responded to this question, representing about 5% of the total survey responses. Some of these students provided answers that included more than one difficulty related to the question, contributing to multiple themes.

The results in Table 5.9 show that the most frequently reported difficulties, mentioned by 50% of respondents, include finding or understanding translations of technical terms in their native language, as well as understanding learning materials, assignments, and exams provided in English. Additionally, 33% of responses reported issues related to general English comprehension. Finally, 17% of students who answered this question stated that they struggle to understand programming concepts due to the use of English in class.

Themes	<i>n</i> of responses contributing to the theme (%)
Difficulty finding or understanding translations/ meanings in native language	3 (50%)
Difficulty understanding learning materials/ assignments/ exams	3 (50%)
General English language difficulty	2 (33%)
Difficulty understanding programming concepts in English	1 (17%)

Table 5.9: Difficulties using English in CS1 class - Case Study Two (Saudi Arabia)

5.4.2.2 The difficulties students face when using English-based programming languages.

In this survey question, students were asked to rate the difficulty level of various tasks related to learning and using English-based programming languages. The question included nine items, with students selecting one of five options for each: '*very easy*', '*easy*', '*neutral*', '*difficult*', '*very difficult*'. The results in Figure 5.6 indicate that the most challenging tasks for students when learning English-based programming languages include understanding and using error messages, as well as intuitively understanding the meaning and functionality of programming language keywords. Between 48% and 50% of surveyed students found these tasks either difficult or very difficult. Additionally, 41% of students reported difficulty in understanding the relationship between

programming language keywords and their meaning or functionality. Furthermore, 30% to 35% of students found it difficult to understand or use IDE hints, tips, help commands, and error messages written in English.

Only about 10% to 20% of students reported difficulties in remembering the correct spelling of programming language keywords, reading programming code, and using the English interface of the programming software (IDE). The least challenging task for students was writing programming code using English letters, with fewer than 5% experiencing difficulty and approximately 55% finding it easy.

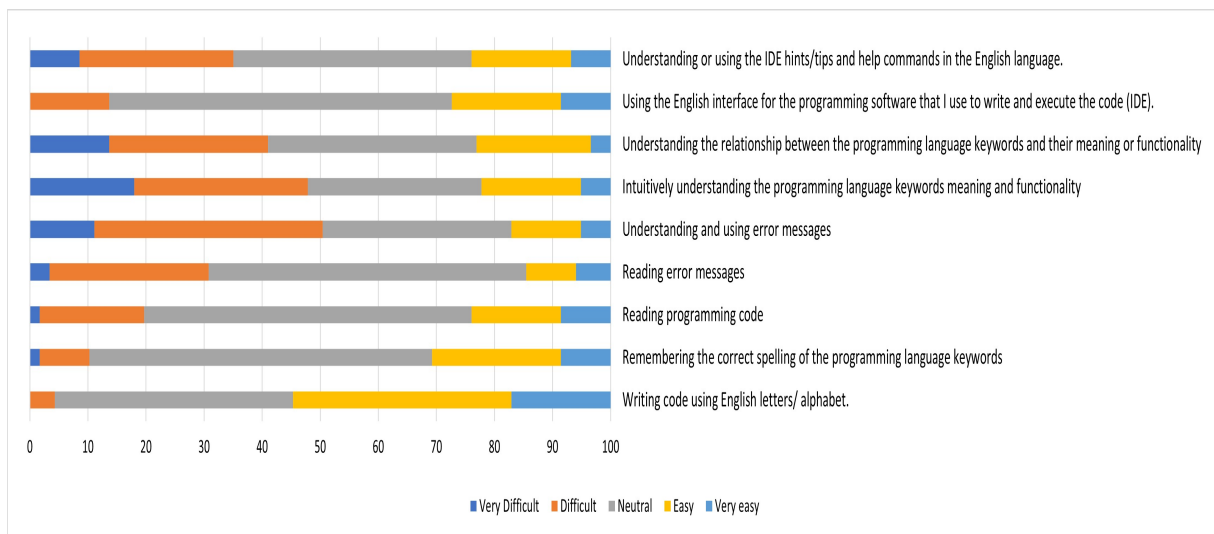
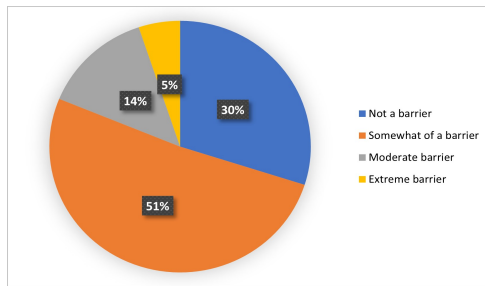


Figure 5.6: Difficulties with using English-based programming languages - Case Study Two (Saudi Arabia)

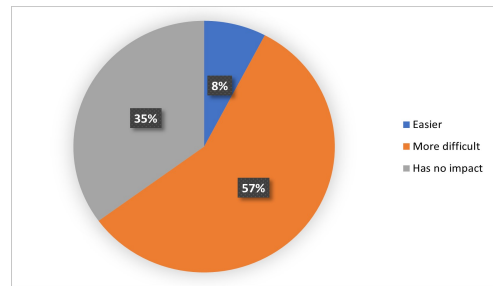
The follow-up open-ended question invited students to add or explain any other difficulties they face when learning English-based programming languages. Five responses were collected; however, only three directly addressed the question. Two responses (40%) mentioned difficulties related to the use of English in class, which were already discussed in the previous open-ended question. Two respondents (40%) reported confusion and difficulty distinguishing similar programming keywords and their functionalities. One student (20%) stated difficulty in understanding and remembering programming language syntax.

5.4.2.3 To what extent do students consider the use of the English language as a barrier to learning programming?

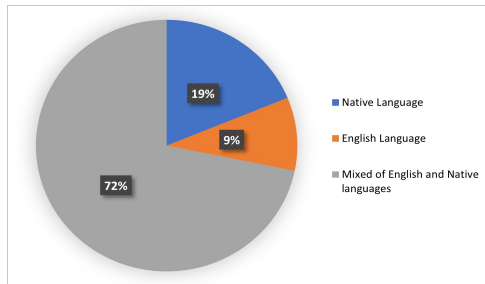
The result shows that 51% of students find the use of English in the CS1 course somewhat of a barrier to learning programming (Figure 5.7a). The same factor represents a



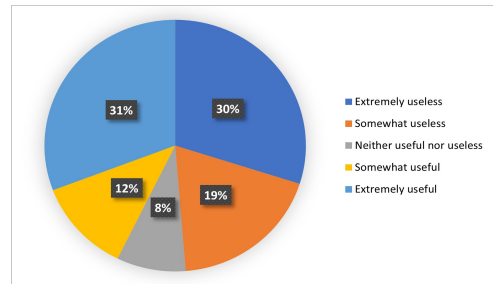
(a) Is English a barrier for learning



(b) Impact of English on course difficulty



(c) Preferred language(s) to use in the course



(d) Perceived usefulness of using English

Figure 5.7: Students' perspectives about using English in CS1 course - Case Study Two (Saudi Arabia)

moderate barrier for 14% of students and an extreme barrier for 5% of students. For 30% of students, using English in the CS1 class is not a barrier at all.

5.4.2.4 What is the impact of using English on the level of the CS1 course difficulty as perceived by students?

The result in Figure 5.7b shows that more than half of the surveyed students (57%) believe that the use of English in the CS1 course makes it more difficult. On the other hand, only 8% think the use of English makes the course easier. 35% of the students said that the use of English in the CS1 course has no impact on the course difficulty.

5.4.2.5 What language(s) do students prefer to use to study programming?

Figure 5.7c shows the responses of the language(s) students prefer to use in the CS1 course. A mix of English and their native languages was the most preferred option for the students (72%). The second most preferred option was the students' native language, representing 19% of the results. Lastly, only 9% would like to continue using only English in the CS1 course.

5.4.2.6 Do students consider the use of English in this course as beneficial for their future study and career?

The result in Figure 5.7d shows that 31% of the students believe that the use of English in the CS1 course is extremely beneficial for their future studies and career. Almost the same number of students (30%) think the opposite and believe it is extremely useless. 12% of the students think it is somewhat useful for their future studies and career, whereas 19% think it is somewhat useless. Only 8% of the students have a neutral opinion about the usefulness of using English in the CS1 course.

5.4.3 Correlation Analysis Results

The scatterplot in Figure 5.8 explores the possible relationships between the dependent and independent variables. The first row (A) in the scatterplot shows a possible positive association between the students' English proficiency level, as measured by the SET, and their academic performance in CS1 courses. The three academic performance indicators (i.e. assignments score, final exam score, and overall CS1 course score) appear to be correlated with the students' English proficiency level.

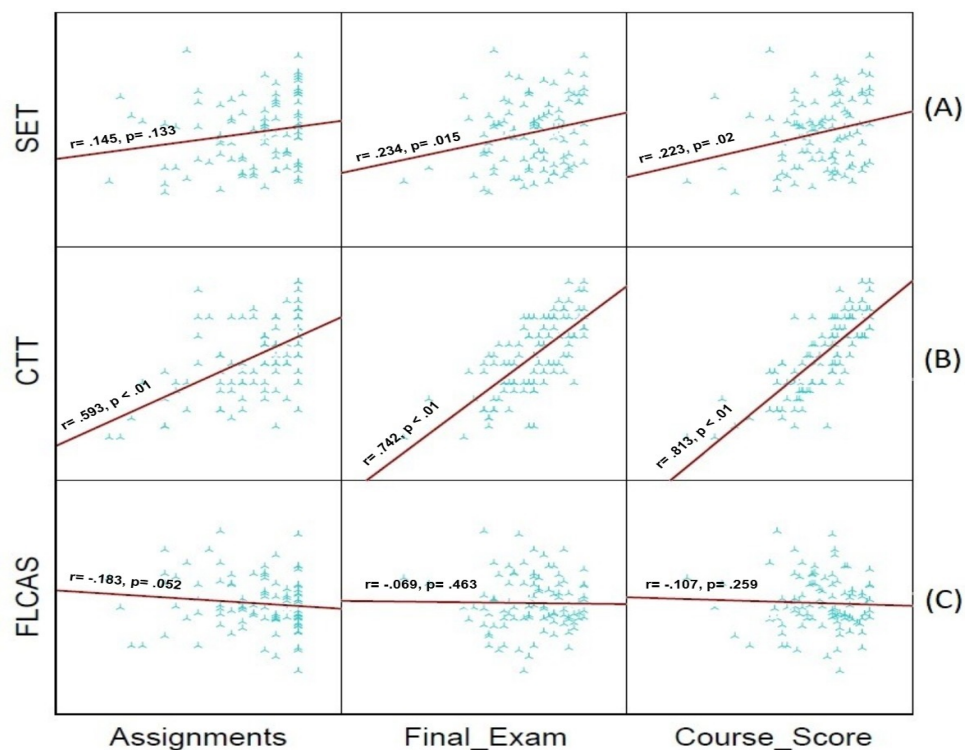


Figure 5.8: Scatterplots matrix - Case Study Two (Saudi Arabia)

The second row (B) shows a high possibility of a positive association between the students' level of knowledge of computer terminology, as measured by the CTT, and their

academic performance in CS1 courses. The students' scores in the final exam appeared to be more correlated with the CTT scores than the assignment scores.

The third row (C) explores the associations between the level of foreign language classroom anxiety, as measured by the FLCAS, and academic performance in CS1 courses. A negative relationship between FLCAS and the three performance indicators is possible; however, it appears to be very weak.

Table 5.10 shows the result of the Pearson correlation (r), which was calculated to test the significance of the observed associations between SET, CTT, FLCAS, and the three academic performance indicators in the CS1 course. The table presents three important numbers: including r , p -value, and the number of cases used in each correlation analysis. Similar to Case Study One, colour codes were used to highlight the strength of the correlations; the lighter the colour, the weaker the correlation.

The correlation results show a significant low to moderate positive correlation between English proficiency level, as measured by the SET, and academic performance in the final exams and overall CS1 course score ($r=.234$, $r=.223$, $p<0.05$, respectively). No significant correlation was detected between SET and students' academic performance in assignments.

The CTT shows a high positive correlation with all the academic performance indicators in the CS1 course. This correlation was stronger with the final exam than with the assignment ($r=.742$, $r=.593$, $p<0.01$, respectively). The highest recorded correlation was with the overall CS1 course score ($r=.813$, $p<0.01$). The positive correlation suggests that higher performance in the English proficiency test or CTT is correlated with higher academic performance in the CS1 course.

On the other hand, FLCAS shows a negative correlation with students' academic performance; however, the correlations are low and not significant.

	Assignments Score			Final Exam Score			Overall Course Score		
	r	p	n	r	p	n	r	p	n
SET	0.145	0.133	108	0.234*	0.015	108	0.223*	0.02	108
CTT	0.593**	<.001	115	0.742**	<.001	115	0.813**	<.001	115
FLCAS	-0.183**	0.052	114	-0.069	0.463	114	-0.107	0.259	114

** Correlation is significant at the 0.01 level.
* Correlation is significant at the 0.05 level.

Correlation Strength: Weak ($-.20 \leq r \leq .20$) moderate ($.20 < r \leq .50$) High ($r > .50$)

Table 5.10: Correlation Results - Case Study Two (Saudi Arabia)

5.4.4 Multiple Linear Regression Analysis Results

As explained in Case Study One (China), the main goal of conducting multiple linear regression analysis is to identify the best predictor for students' academic performance in the CS1 course using the three independent variables explored in this research. The summary of the linear regression model in Table 5.11 shows that 63.7% ($r^2=.637$) of the variance in students' academic performance in the CS1 course can be explained by the independent variables (i.e., SET, CTT, and FLCAS). This result is satisfactory for proceeding with the linear regression analysis results.

Country	Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
SA	1	0.798 ^a	0.637	0.625	8.902

a. Predictors: (Constant), SET, FLCAS, CTT
b. Dependent Variable: Course_Grade

Table 5.11: Model Summary^b - Case Study Two (Saudi Arabia)

Table 5.12 shows that this prediction model can be reliably used to predict students' academic performance in the CS1 course using the group of three independent variables (Sig. < 0.01). To determine which of the independent variables is the best predictor of students' performance, the coefficient table should be analysed.

Country	Model		Sum of Squares	df	Mean Square	F	Sig.
SA	1	Regression	12528.94	3	4176.312	52.696	.000 ^b
		Residual	7132.809	90	79.253		
		Total	19661.75	93			

a. Dependent Variable: Course_Grade
b. Predictors: (Constant), English_Proficiency, FLCAS, CTT

Table 5.12: ANOVA^a - Case Study Two (Saudia Arabia)

Table 5.13 shows the ability of each of the three independent variables to predict students' academic performance in the CS1 course. *P*-value that indicates the significance of the predictor in predicting students' academic performance is shown in the last column (Sig). This value should be < 0.05 to reject the null hypothesis and accept the independent variable as a reliable predictor. CTT is the only independent variable that shows a *p*-value < 0.05. For this variable, the B value is 0.803, meaning that 0.8 points are predicted to increase in students' overall CS1 course score for every point increase in the CTT score.

Similar to the results of Case Study One (China), the results of the linear regression in this case study show that CTT is the only independent variable that can predict stu-

Country		Unstandardized	Coeffi-	Standardized	Coef-	t	Sig.
		B	Std. Error	Beta			
SA	(Constant)	27.64	6.068			4.555	<.001
	CTT	0.803	0.067	0.771		11.939	<.001
	FLCAS	-0.028	0.038	-0.047		-0.735	0.464
	SET	0.131	0.077	0.11		1.701	0.092

a. Dependent Variable: Course_Grade

Table 5.13: Multiple linear regression coefficients^a - Case Study Two (Saudi Arabia)

dents' academic performance in the CS1 course. Neither SET nor FLCAS has the ability to reliably predict students' academic performance in the CS1 course. Accordingly, testing the relationship between the two variables, i.e. CTT and overall CS1 course scores, can be done more appropriately using simple linear regression. The result of the simple linear regression shows r^2 equal to 0.659, which means that 66% of the variability in the overall CS1 course score can be predicted from the relationship with CTT. The B value is equal to 0.836, which means that for every point increase in the CTT score, an increase of 0.8 units in the overall CS1 course score can be predicted, which is very similar to the result obtained from the multiple linear regression.

5.5 Cross-Case Analysis

5.5.1 Participants' Background

Table 5.14 shows a comparison of the background of participants in both case studies based on the background survey, which looks very similar. A significant difference appeared in the number of participants who have completed a subject on CS basics in high school, in favour of students in Case Study Two (Saudi Arabia). The level of the CS subject they completed is not specified; however, it is not expected to have a significant impact on the research results. The CS basics subject provided in schools usually covers basic knowledge and applications and is taught in the students' native languages. It is more likely that the CS basics they completed in high school did not include programming, as the majority (85%) said that the CS1 course they were attending was the first formal programming course.

The main differences that might have been reflected in the results are the language the teachers use and the language students can use in class. In Case Study One (China),

	Case Study One (China)	Case Study Two (Saudi Arabia)
Number of participants in the background survey	276	117
Non-native English-speaking students	✓	✓
The participants' native language and English language share the same alphabet or letters.	X	X
Aged between 18- 23	98%	99%
Gender (F= Female, M=Male, U=Undetermined)	25% F, 74% M, 1% U	32% F, 68% M
First-year university level students	✓	✓
CS1 course is the first programming course in the programme.	✓	✓
EMI CS1 course (provided in English).	✓	✓
Students are studying English course simultaneously with CS1 course.	✓	✓
The CS1 courses are provided in a non-English speaking country	✓	✓
Participants never attended English courses in any of the English-speaking countries.	96%	96%
Participants went to high schools that do not apply EMI.	89.50%	98%
Students attended maths, science, English courses in high school.	97.5%, 94%, 97%; respectively	97% maths and science, 100% English
Attended a computer science basic course in high school	44%	97%
First formal programming courses	85%	85%
Self-learning programming prior the CS1 course	17%	13%
Participants rated themselves as having no experience or being novices in programming.	77%	86%
CS1 Course instructors are native English speakers	✓	X
Students can use native language in class	48%	100%
Teacher use only English in class	92%	3%
Teacher can provide explanations in native language for exams and assignments	4%	40%
Books and learning materials are provided only in English	97%	100%

Table 5.14: A comparison of participants' backgrounds

only 48% of the students said that they can use their native language in class, compared to 100% in Case Study Two (Saudi Arabia). This question did not specify with whom the students can use their native language; thus, it could be with the teacher, teaching assistant, or other students in the class.

Moreover, in Case Study One (China), 92% of the students stated that their teachers use only English in class, compared to 3% in Case Study Two (Saudi Arabia). However, in Case Study Two, 86% of the students said that their teachers use a mix of English and native languages, and 11% of the students said that teachers use only their native language. The teachers in Case Study One (China) are all native English speakers and do not speak the students' native language. However, 8% of the students said that the teacher uses a mix of English and their native language in class. This is probably because some students in Case Study One (China) had teaching assistants who spoke their native language. It is possible that the teaching assistants communicated with some students in their native language.

All assignments and exams in the CS1 courses included in this study were provided in English. However, the students were also asked if they were able to receive explanations for assignments and exam questions in their native language from their teachers. In Case Study One (China), only 4% said that they were able to receive this type of explanation, whereas 40% of students in Case Study Two (Saudi Arabia) said so.

The similarities between the students in both case studies include age group, majority male, and all NNES students. They were studying English simultaneously with the CS1 course, and the majority had never attended English courses in English-speaking countries in the past. The majority of the students in both case studies attended regular high schools (not EMI) and completed main science subjects, including maths. Also, almost all students agreed that the books and learning materials used in the CS1 courses were only provided in English.

The level of programming experience they had prior to the CS1 course was closely similar based on a self-reporting question. However, higher numbers of novices and students with no experience in programming were expected in this type of course. One possible explanation is that the timing of the survey at the end of the semester might have impacted the results. Thus, some students might have misunderstood the question and rated their level of experience at the time of the survey.

These similarities and differences can help explain some of the patterns that appeared in the correlation results. They might also explain what could be responsible for these patterns.

5.5.2 The Students' Perspectives

The survey explored the students' perspectives on using English in CS1 courses and focused on the difficulties they face. Students in both case studies reported difficulties

asking and answering questions (Table 5.15). This might have resulted in difficulty understanding programming concepts when explained in English, which both groups struggle with. Difficulties with understanding computer terminology affect students in Case Study One (China) more frequently, while students in Case Study Two (Saudi Arabia) face more English language difficulties, including writing and listening problems. In the open-ended question, all students reported general English language difficulty. Students in Case Study One (China) reported difficulties with technical terms, while students in Case Study Two (Saudi Arabia) reported difficulty finding translations for those terms. Students in Case Study Two (Saudi Arabia) also reported difficulties understanding learning materials provided in English.

The survey also asked the students about the difficulties they face when using English-based programming languages. In the results, all students face difficulties when dealing with error messages and IDE hints and tips. Students in Case Study One (China) also face difficulties dealing with the IDE English interface, while students in Case Study Two (Saudi Arabia) face more difficulties with the programming language keywords and their functionality. In the open-ended questions, comprehending programming code and dealing with error messages, as well as problem-solving difficulties, were reported by students in Case Study One (China). On the other hand, students in Case Study Two (Saudi Arabia) reported difficulties related to distinguishing programming language keywords.

The survey also asked the students which language or languages they would prefer to use in the CS1 course. Surprisingly, the results from both case studies were exactly the same, with the majority of the surveyed students preferring to be able to use a mix of English and their native language.

This survey included a question about the students' opinions on to what extent they consider English a barrier to learning programming. The results of both case studies were very close in this matter, except for the students who do not consider English a barrier at all. The number of students who did not consider English a barrier at all was higher in Case Study Two (Saudi Arabia). This might be a result of being able to use their native language in class, as well as receiving explanations for exams and assignments in their native language.

The survey also asked the students if the use of English makes the CS1 course easier, more difficult, or has no impact. A large number of students in both groups believed that English makes the CS1 course more difficult; however, the percentage of students who think so is higher in Case Study One (China). Again, the ability of the students in Case Study Two (Saudi Arabia) to use their native language might have influenced their opinion.

The students in the two case studies had noticeably different views on how useful English is in their CS1 course, and these differences seem to reflect the unique circumstances of each case (Table 5.15). In Case Study One (China), 52% of the surveyed students believed that using English is '*extremely beneficial*' for their future studies and careers. On the contrary, in Case Study Two (Saudi Arabia), only 31% shared that perspective, with a difference of about 20%.

This big contrast in opinions might be explained by the differences in the educational and economic environments of the two countries. In China, for example, the rapid global integration of the tech industry and the importance of English in international academic settings could make students see English as an essential skill for success. Notably, the students in Case Study One (China) are enrolled in bachelor's degree programmes at an international college; these programmes are a collaboration between a Chinese university and an Irish university. Therefore, it is reasonable to expect that these students are preparing for future careers in international and multicultural working environments, where proficiency in English is essential. In contrast, students in Saudi Arabia who attend a local university (Case Study Two) might anticipate working locally in fields where English plays a less central role, which could make them view the use of English in their CS1 course as less valuable. For example, one of the most attractive jobs for CS graduates in Saudi Arabia is teaching in public schools, ranging from elementary to high school levels, where Arabic, not English, is the primary medium of instruction. This perspective may also help explain why some Saudi students even described using English in the course as '*extremely useless*', as English proficiency is not a requirement for a teaching position.

Differences in perception highlight how cultural backgrounds, economic realities, and career aspirations can influence students' attitudes toward English-medium instruction in CS. It is clear that students' perspectives are shaped by more than just the language itself or the specifics of the course content; they are also affected by the broader contexts in which these students live and study.

5.5.3 Correlation Results Comparison

5.5.3.1 Standard English Test (SET)

Table 5.16 shows the correlation results between the dependent variables and the SET for the two case studies, alongside z statistics values. Low positive correlations were found between SET and the students' academic performance in CS1 final exams and overall CS1 course scores in both case studies. A low positive correlation is also noticed

	Case Study One (China)	Case study Two (Saudi Arabia)
Top difficulties with using English in CS1 class (reported by over 25% of the participants)	Asking or answering questions in English, Understanding programming concepts when explained in English, Understanding some English computer terminology used in the class. <i>Open-ended question results:</i> General English comprehension, Difficulties with the computer or proficient terminology.	Writing or explaining answers in English, Asking or answering questions in English, Following up with the teacher when speaking English, Understanding programming concepts explained in English. <i>Open-ended question results:</i> Finding or understanding translations of technical terms, Understanding learning materials, assignments and exams provided in English, General English comprehension.
Top difficulties with learning English-based programming languages (reported by over 25% of the participants)	Reading error messages, Understanding and using error messages, Using the English interface for the programming software used for coding (IDE), Understanding or using the ide hints or tips and help commands in the English interface. <i>Open-ended question results:</i> Reading and comprehending programming code, Using and understanding error messages, Problem-solving and computational thinking difficulties.	Understanding and using error messages, Intuitively understanding the programming language keywords' meaning and functionality, Understanding the relationship between the programming language keywords and their meaning or functionality, Understanding or using the ide hints/tips and help commands in the English language, Reading error messages. <i>Open-ended question results:</i> Confusion and difficulties distinguishing similar programming keywords and their functionalities.
The top language(s) students prefer to use in CS1 class.	Mix of English and their native language (72%).	Mix of English and their native language (72%).
The level of the barrier English represents to learning programming	16% is not a barrier at all 56% somewhat of a barrier 20% moderate barrier 5% extreme barrier	30% is not a barrier at all, 51% somewhat of a barrier, 14% moderate barrier, 5% extreme barrier.
Do the students think the use of English has an impact on the CS1 course?	73% said it makes CS1 course more difficult. 23% No impact 4% Easier	57% said it makes CS1 course more difficult. 35% No impact 8% Easier
What students think about the usefulness of using English in the CS1 course	52% believe that using English in the CS1 course is extremely beneficial for their future studies or career	31% of the students believe that the use of English in CS1 course is extremely beneficial for their future study, 30% think it is extremely useless.

Table 5.15: Comparing Students' perspectives.

in Case Study One (China) between SET and academic performance in assignments; however, no similar correlation was found in Case Study Two (Saudi Arabia). The absence of any significant values in the z statistics means that there is no significant difference in any of the correlation results. Thus, it is possible to say that SET has a low positive correlation with the students' academic performance in the CS1 course in general.

	Case Study One (China)	Case Study Two (Saudi Arabia)	z -value
	r	r	
Assignment	0.207**	0.145	0.308
Final Exam	0.160*	0.234*	-0.377
CS1 Score	0.211**	0.223*	-0.063

** Correlation is significant at the 0.01 level

* Correlation is significant at the 0.05 level

Table 5.16: SET Correlation Comparison

5.5.3.2 Computer Terminology Test (CTT)

Table 5.17 shows the correlation results between CTT and different dependent variables for both case studies, as well as z statistic values. High positive correlations were identified between CTT and the students' academic performance in CS1 assignments and overall CS1 course scores in both case studies. The correlation between CTT and academic performance on the final exam is moderate in Case Study One (China) and high in Case Study Two (Saudi Arabia). Based on the z -value results, the difference between the correlations of CTT and assignments is not significant. However, the difference is significant in the case of the final exam and the overall CS1 course score. The difference is higher in the case of the final exam and lower in the case of the overall CS1 course score.

In general, it can be said that the correlation between CTT and the students' academic performance in assignments is strong and stable. However, the strength of the correlation between CTT and the final exam and overall CS1 course scores changed significantly based on the case studies. This difference might be related to the nature and circumstances of the final exams, but the exact reason is hard to identify.

	Case Study One (China)	Case Study Two (Saudi Arabia)	<i>z</i> -value
	<i>r</i>	<i>r</i>	
Assignment	0.566**	0.593**	-0.267
Final Exam	0.367**	0.742**	-3.733**
CS1 Score	0.699**	0.813**	-1.977*

** Correlation is significant at the 0.01 level

* Correlation is significant at the 0.05 level

Table 5.17: CTT Correlation Comparison

5.5.3.3 Foreign Language Classroom Anxiety Scale (FLCAS)

Table 5.18 presents the correlation results between FLCAS and the students' academic performance in the assignment, final exam, and overall CS1 course score, alongside *z* statistic values. In general, FLCAS shows negative correlations with the students' academic performance in the CS1 courses. However, none of these correlations were significant except for the academic performance in the assignment for Case Study One (China), which is still a weak correlation. The results of *z* statistics show no significant difference between the correlation results. Thus, it is possible to say that there is no significant correlation between the overall FLCAS level and students' academic performance in general.

Although the main focus of this thesis was on the overall FLCA level, it was interesting to look at different underlying factors related to FLCA that can be measured by the scale used. Therefore, a more detailed factors analysis was conducted on the data collected using the FLCAS in the academic year of 20/21. The results showed that factors related to speaking had more correlation with students' academic performance in assignments than other studied factors. These two factors include uncomfortable-ness speaking with native speakers and a lack of confidence in speaking English in class. This might have impacted the students' willingness to ask for more clarification about assignments. The full result was published in a conference research paper (see Appendix G), and the following is a summary of the main findings.

The research presented in that paper explored the correlation between the students' academic performance in the CS1 course and the overall FLCAS as well as some of its underlying related factors. Six of the related factors that were mentioned in previous studies in the area were investigated, including (1) uncomfortableness speaking with native speakers, (2) lack of confidence in speaking English in class, (3) speech anxiety, (4) fear of negative evaluation, (5) fear of failing the class as a consequence of not being

	Case Study One (China)	Case Study Two (Saudi Arabia)	<i>z</i> -value
	<i>r</i>	<i>r</i>	
Assignment	-0.193**	-0.183	-0.037
Final Exam	0.008	-0.069	0.325
CS1 Score	-0.127	-0.107	-0.078

** Correlation is significant at the 0.01 level

Table 5.18: FLCAS Correlation Comparison

good at English, and (6) anxiety about not understanding everything taught in English. Each factor was measured by three to twelve questions from the main FLCAS. These underlying factors, as well as the items used to measure them, were previously identified by other researchers [242, 249, 246, 243, 248, 247, 245].

The data used in this investigation were acquired from the first data collection during the academic year of 20/21 of the two case studies. It includes the data collected from 160 students: 113 students in Case Study One (China) and 47 students in Case Study Two (Saudi Arabia). The results presented in this paper were based on the analysis of all data as a single group, as well as at the level of groups.

When the correlation analysis was conducted between the overall FLCA level and all participants' academic performance in the CS1 course, a significant but low negative correlation was detected with assignment scores only. After repeating the correlation analysis on a group level, no significant correlation was detected. The interesting results appeared when the groups were divided by gender, not by country. Male students showed a significant negative correlation between the overall FLCA level and the assignment score, despite the average FLCA level being slightly higher for female students.

Moving to the six related factors, the analysis was also conducted on all participants' data and at the group level. In the results, five of the six factors (factors number: 1, 2, 3, 5, and 6) showed a significant but low negative correlation with the assignment score. At the group level, in Case Study One (China), only two factors (factors 1 and 2) showed a significant correlation with students' academic performance in programming assignments, and no significant correlations were detected in Case Study Two (Saudi Arabia). The paper includes more detailed results and discusses these findings in light of previous work in the area [251].

	Multiple linear regression		Simple linear regression	
	r^2	B	r^2	B
Case Study One (China)	0.566	0.656	0.488	0.708
Case Study Two (Saudi Arabia)	0.637	0.803	0.659	0.836

Table 5.19: Linear Regression Comparison

5.5.4 Multiple Linear Regression Results Comparison

The model summary for Case Study One (China) reveals that the model, which includes the three independent variables (SET, CTT, and FLCAS), can explain 56.6% ($r^2 = .566$) of the variance in the students' academic performance in the CS1 course. This was slightly higher in Case Study Two (Saudi Arabia), where the same model with the same independent variables is able to explain 63.7% ($r^2 = .637$) of the variance in the students' academic performance in the CS1 course. By examining the ability of each individual variable to predict academic performance in the CS1 course, CTT was the only independent variable that could reliably be used as a predictor in both case studies. In Case Study One (China), a 0.656-point increase in the overall CS1 course score is expected for every one point increase in the CTT score. In Case Study Two (Saudi Arabia), this increase is equal to 0.803 for every point increase in CTT. This difference is a result of the lower correlation that Case Study One (China) showed between CTT and final exams. As mentioned before, this might be impacted by differences in the nature and circumstances related to the final exam sitting.

In both case studies, only one independent variable showed significant results, which is why simple regression was conducted. The simple regression analysis included CTT as an independent variable and CS1 course score as a dependent variable. The result in Table 5.19 shows that r^2 for Case Study One (China) decreased from 0.566 in the multiple linear regression to 0.488 in the simple linear regression results. However, the B value for Case Study One (China) increased, becoming closer to the B value of Case Study Two (Saudi Arabia). For Case Study Two, the figures of the simple linear regression are not significantly different from the multiple linear regression results. This means that using only CTT can explain between 49% and 66% of the variability in the overall CS1 scores. Moreover, between 0.7 and 0.8 points are expected to increase in the overall CS1 course score for every point increase in the CTT score.

5.6 Research Hypotheses Testing

The multiple linear regression results can be used to test the research hypotheses, as explained by McNeil et al. [263]. The most important value for hypotheses testing is the *Sig.* value, which should be less than the tolerable level of significance chosen for the study. In this research, the *Sig.* value should be below 0.05 for a 95% confidence interval. Based on the significant value for each factor under investigation, the null hypothesis is either rejected or not rejected. The null hypothesis is not rejected if the *Sig.* is > 0.05 , while if *Sig.* is < 0.05 , the null hypothesis is rejected and the alternative hypothesis is accepted. In other words, if the significant value is less than 0.05, then there is an impact; if it is greater than 0.05, then there is no impact (i.e., the impact of the independent variable on the dependent variable).

The following are the three research hypotheses, as presented in Chapter One: section 1.5, along with the alternative hypotheses:

1. **Null hypothesis H01:** The level of English language proficiency does not impact NNES students' academic performance in CS1 courses.

Alternate Hypothesis HA1: The level of English language proficiency impacts NNES students' academic performance in CS1 courses.

2. **Null hypothesis H02:** The NNES students' level of knowledge in English computer terminology does not impact their academic performance in the CS1 course.

Alternate hypothesis HA2: The NNES students' level of knowledge in English computer terminology impacts their academic performance in CS1 courses.

3. **Null hypothesis H03:** Foreign language classroom anxiety level does not impact NNES students' academic performance in CS1 courses.

Alternate hypothesis HA3: Foreign language classroom anxiety level impacts NNES students' academic performance in CS1 courses.

Based on the results, there is a correlation between NNES students' academic performance in the CS1 course and the three studied factors, although the strength of these correlations varies. The English proficiency level showed medium to low significant correlations across all course aspects in both case studies, and these correlations were mostly stable. However, the multiple linear regression results did not indicate any significant effect. Consequently, the decision to accept or reject the first hypothesis of this study remains inconclusive.

Conversely, the NNES students' knowledge level of English computer terminology has an impact on their academic performance in the CS1 course. This is supported by

the correlation and multiple linear regression results, which suggest that the second hypothesis (HA2) of this study can be confidently accepted.

The FLCAS level clearly has no noticeable impact, as the correlation was significant, yet weak, with assignment scores for only one of the case studies. The multiple linear regression results also support the acceptance of the null hypothesis (H03) related to FLCA, rejecting the impact of this factor on NNES students' academic performance in the EMI CS1 course. Table 5.20 shows the details of the hypotheses testing results with the interpretation.

Independent variable	Sig. Value		Hypothesis Testing Result at 95% confidence interval	Interpretation
	Case Study One (China)	Case Study Two (Saudi Arabia)		
English language proficiency measured by SET	0.848	0.092	Null Hypothesis cannot be rejected in both case studies	Based on this result, the level of English language proficiency might not impact NNES students' academic performance in the CS1 course. However, the correlation figures presented in Table 5.16 suggest the opposite. Therefore, this hypothesis cannot be conclusively rejected or accepted.
Level of knowledge in English computer terminology measured by CTT	<.001	<.001	Null Hypothesis can be rejected in both case studies	The NNES students' level of knowledge in English computer terminology impacts their academic performance in the CS1 course.
Foreign language classroom anxiety level measured by FLCAS	0.132	0.464	Null Hypothesis cannot be rejected in both case studies	Foreign language classroom anxiety level does not impact NNES academic performance in the CS1 course.

Table 5.20: Hypotheses Testing Results

5.7 Conclusion

This chapter presented detailed analysis results of the two case studies conducted in this thesis. For each case study, participants' backgrounds and perspectives on the different aspects of the CS1 course were presented first to provide a foundation for the correlation and multiple linear regression results that were presented later. Following that, a cross-case analysis was presented to show the similarities and differences between the results of the two case studies. Finally, the results of hypotheses testing, based on the multiple case studies, were presented. The discussion of these findings in light of previous literature and the answers to the thesis' questions are presented in chapter 7.

The next chapter (chapter 6) presents the development of an online managed and peer-reviewed terminology translation activity that was introduced and evaluated in a CS1 course. The highly significant correlation between the students' knowledge of computer terminology and their academic performance in the CS1 course was the main motivation for the development of this learning activity.

MANAGED AND PEER-REVIEWED TRANSLATION ACTIVITY FOR COMPUTER TERMINOLOGY

6.1 Introduction and Motivation

The results presented in the previous chapter show a significant correlation between students' performance in CTT and their academic performance in CS1 courses. The impact of students' knowledge of computer terminology was not entirely surprising, as it has been mentioned in some previous studies [32, 33]. The results of this research and previous studies highlight the need for learning strategies and tools that can help NNES students improve their knowledge of such terminology, which is expected to positively impact their learning of programming.

One proposed approach is the provision of extra classes to pre-teach CS vocabulary to students [76]. However, this requires a significant amount of preparation and extra class hours from both teachers and students. Additionally, these extra classes do not involve active learning, which is recommended by many educators to enhance the benefits of the learning process [211, 212]. Other proposed approaches include developing a mobile application [209, 205] and an electronic game to teach CS terminology [208, 207]. These applications are limited, as they require considerable effort from experts fluent in both English and the learners' native language to make them viable for a specific language. Unlike previous approaches, the purpose of the proposed translation activity is to be lightweight, easy to implement in class, require minimal preparation and resources, and promote active learning and engagement.

The main objective of designing and implementing this activity is to help NNES students improve their knowledge of computer terminology by encouraging them to find

and evaluate translations of selected terms from the course they are attending. The learning goals of this activity, along with the designed tool, include the following:

- Improve knowledge of computer terminology included in the course by asking students to provide suitable translations for the selected terms.
- Enhance critical thinking and problem-solving skills by asking students to review and vote on the translations of their peers.
- Increase learning motivation and engagement in CS1 courses through active peer review and by engaging all students in the production of the course glossary.

In foreign language education research, translation activities have been found to be beneficial for terminology learning, including retention and accuracy of learned terms [310]. Finding translations is one of the main difficulties that NNES students face in EMI courses, and it consumes a lot of their time while studying [311]. The aim of this activity is to use translation as an approach to enhance students' understanding and knowledge of the computer terminology required in CS1 courses. It also seeks to make use of the time NNES students typically spend outside the classroom searching for translations and the knowledge they gain from this process. This is expected to help students feel that the time they spend searching for and learning new terminology related to the course is valuable and appreciated, rather than just an extra burden.

Both the act of creating translations of important terms from class and reviewing the translations of their peers are fundamental to this activity. Besides its importance in enhancing students' knowledge and skill-building in the specific-domain subject [312, 313], peer review can also help develop critical thinking and problem-solving skills [314, 315]. Peer review activities can also assist students in improving their ability to judge other people's work more objectively and help them become more open to accepting and benefiting from feedback [316].

The electronic voting system (EVS) is widely used in higher education to increase student engagement and promote active learning, which inspired the idea of voting in the proposed translation activity [317]. EVS is applied in various ways, but it is mostly used during lectures to collect students' feedback on the learning process, revise previous lectures to identify misunderstandings, and support exam preparation [317]. However, the voting in this activity does not aim to diagnose students' understanding of the terminology. Instead, it is designed to motivate them to collaboratively build their own glossary for the course using the material they produce. This is expected to positively influence the students' engagement in the activity.

The main feature of EVS is the anonymous nature of responses, which encourages more students to engage in such activities [317]. Similarly, peer review is proven to be more effective when students can give their feedback anonymously [318]. Thus, the peer review and voting responses in the translation activity were designed to be anonymous to promote these benefits.

This chapter presents the design and implementation of the managed and peer-reviewed terminology translation activity. A quasi-experimental research approach was adopted to investigate the effectiveness of this learning activity. Moreover, the results of the students' feedback survey regarding this activity are presented at the end of the chapter¹.

6.2 The Activity Design

The terminology translation activity applied in this experiment includes three sub-activities: (1) translation, (2) review, and (3) voting. Figure 6.1 summarises the sub-activities included in the terminology translation activity.

In the first sub-activity, the teacher will assign one term to each student to translate and ask them to provide a short description or explanation for it in their native language. A list of terms will be prepared in advance to be used in this activity. No English definition of the term will be provided to prevent students from using a direct translation of any definition supplied. The students will be asked to keep their translation to a one-paragraph length; however, the text box in the tool will accept a significant length of text. The students will be allowed to add and edit their answers on the application until the deadline, at which point they must submit the final copy of the translation, with no editing allowed after submission is completed.

The second sub-activity is the review, which includes a review of the submitted translations by teaching assistants as well as by multiple peers. The teacher will first assign the translations to the teaching assistants for review. They will be asked to select one of the provided options: 'inadequate', 'adequate', 'good', or 'excellent'. A short description of each review option will be provided as follows:

¹Large parts of this chapter were presented in the following published papers:

- Russell, S., Alaofi, S., and Alshaigy, B. 'Translate Together: Managed Translation and Peer-Review.' in *Proceedings of the Innovation and Technology in Computer Science Education Conference (ITiCSE)*, short paper, 2023, 2 pages. [Online].
- Alaofi, S., and Russell, S. 'Improving Knowledge of CS1 Terminology Through a Peer Reviewed Translation Activity: Results and Feedback' in *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE)*, article, 2024, 7 pages. [Online].

- **Inadequate:** The translation was inaccurate or poorly structured. This would include translations that read like they were done by an algorithm.
- **Adequate:** The translation of the term is accurate, but the description is not very detailed or clear.
- **Good:** The translation of the term was good and the description was good too, but was lacking in some detail that would have made it better.
- **Excellent:** The translation of the term and the description are excellent. The description explains this concept really clearly and gives enough detail for it to be understood easily.

Each teaching assistant will be assigned all translations of the same term(s) to review, enabling them to nominate the three to four best translations for each term. After that, the translations will be distributed to students for peer review, where each student will be assigned a number of translations to review. The number of reviews required depends on the number of terms used in the activity. For example, if four terms are used, each student will be required to review three translations. These translations will be selected such that each student reviews a single translation of each of the terms they were not required to translate. As such, all students will encounter all terms either through translation or review.

Students will be asked to select one of the provided review options, which are the same as those used by the teaching assistants, as presented above. They will also be encouraged to add constructive feedback in the form of a comment or explanation for their selected review option, though this is not mandatory.

The third and final sub-activity is voting on the best translation for each term. At this stage, a list of all translations nominated by teaching assistants will be used. Each student will be asked to vote for the best translations of three to five terms. The results of the voting sub-activity will be used to build the course glossary, which will be made available to all students in the class.

These three sub-activities will be repeated throughout the semester, with each set of four terms corresponding to the topics covered in class during that period. More details on the term selection process are provided in subsection 6.4.3.

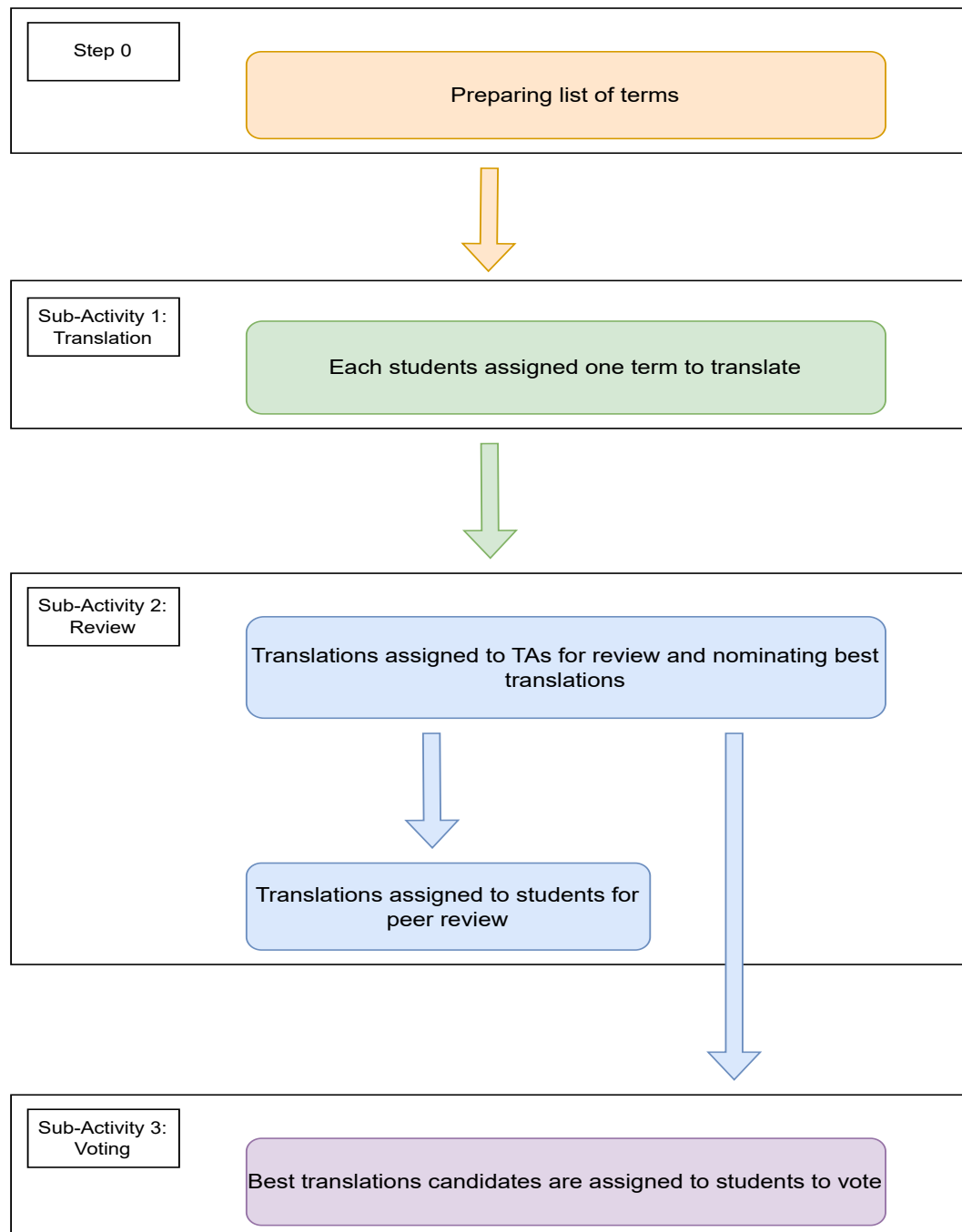


Figure 6.1: Translation activity process

6.3 The Tool Design

Learning management systems (LMSs) play an essential role in higher education and have been used to enhance the learning and teaching experience for students and educators [319]. The importance of adopting technology for online learning and teaching became particularly evident during the COVID-19 pandemic, when many educational institutions worldwide were forced to close their doors [320]. An LMS is a web-based

learning environment that enables teachers and students to carry out most learning and teaching activities online [321].

Despite the many features that LMSs offer, instructors sometimes need to use external applications for specific teaching activities that are not supported by the LMS in use. The use of multiple applications and different platforms for the same course can confuse students and complicate the course management process [322]. A more efficient approach is to use the LMS adopted by the institution to access the necessary external applications, which is possible through learning tool interoperability (LTI) specifications.

LTI can be described simply as a set of standards that enables any external tool or content to be integrated into an LMS [323, 324]. Thus, implementing the translation activity tool as an LTI tool provider makes it possible to integrate it into any compatible LMS, which in this study is Moodle. The primary objective of this tool is to simplify the process of applying and managing the peer-reviewed terminology translation activity for teachers and teaching assistants (TAs) while also making student access to the activity easier.

Using this tool, the teacher will be able to add a list of terms for the translation activity, assign terms to students, and then assign translated terms to TAs for review before distributing them to students for peer review. TAs will also be able to nominate three to four translations of each term for the voting stage while conducting their review.

In this experiment, the teacher does not speak the students' native language, but the TAs do, making their role essential in reviewing students' submissions. This step can be completed by teachers if they share the same native language as their students. Finally, students will vote on the best translation for each term, and the selected translations will be used to build a glossary for students to use in the CS1 course.

The functionalities that this tool provides for each user are detailed below:

- **Teacher:**

- Add lists of terms for the activity,
- Assign terms to students for translation,
- Set deadlines for translation submission,
- Assign submitted translations to TAs for review,
- Assign translations to students for peer review after TAs reviews are completed,

- Assign translations nominated by TAs to students for voting on the best translation.
- **Teaching Assistant (TA):**
 - Review students' translations assigned to them,
 - Nominate three to four translations of the same term for the best translations list.
- **Student:**
 - Add and edit the translation for the assigned term,
 - Submit the translations,
 - Review submitted translations but cannot edit them,
 - View the results of the peer review on their submitted work,
 - View and vote on the best translation.

6.4 Experiment Design and Implementation

6.4.1 Approach

This part of the research adopted a quasi-experimental approach to evaluate the effectiveness of the designed activity and tool. Quasi here means '*resembling*', which clearly suggests that this approach is not a true experiment but rather resembles experimental research [325]. This approach typically lacks one of the main conditions of a true experiment. These conditions are either the randomness of participant selection or the inclusion of a control group [325, 326]. As a result, quasi-experimental research cannot eliminate the impact of confounding variables [325]. It is commonly applied in situations or fields where adopting an experimental approach is challenging, such as educational settings [327, 325]. There are different ways to conduct quasi-experimental research, but the most popular designs are one-group and non-equivalent group designs [325].

The selected approach here is a post-test-only non-equivalent group design, as the participants are not randomly assigned to experimental and control groups by the researcher. The absence of this condition can negatively impact the internal validity of the results if no measures are taken to minimise its effect [325].

The post-test-only approach means that the treatment group will be tested only after the intervention, with no pre-test conducted. The baseline in this case will be the results

of the non-equivalent control group in the same test without receiving any intervention [325]. The main threat of this approach is that any difference detected in the performance of the two groups may be a result of the intervention treatment or may stem from pre-existing differences between the groups, as they are non-equivalent [325, 326].

When adopting this approach, the researcher can maximise the validity of the results by ensuring that the groups are as similar as possible to minimise the impact of most confounding variables [325]. This can be achieved by selecting closely similar classes with teachers of the same gender, using the same teaching methods, etc. [325].

6.4.2 Participants

The experimental group included 90 NNES students enrolled in the CS1 course at Beijing-Dublin International College (BDIC) during the 22/23 academic year. Of these 90 students, 64 completed the CTT, and 67 participated in the feedback survey.

The control group consisted of 178 students who attended the same CS1 course at the same institution during the 20/21 and 21/22 academic years. However, only 83 students from the control group voluntarily participated in the CTT.

The CS1 course for both groups was identical in all respects. The learning materials, topics, teaching methods, and teaching staff remained the same. The course was also delivered online using the same teaching approach for both groups. Additionally, there were no changes in admission policies across those years. The only difference between the courses was the translation activity, which was provided exclusively to the experimental group.

6.4.3 Selected Terms and Timing

The terms chosen for this activity were based on a lexical analysis of the course lecture notes, following a method similar to the generation of the CS1 word list (see Chapter 4: section 4.2). In this experiment, all terms used in the CTT were excluded to prevent bias in the subsequent analysis. Table 6.1 presents the terms used in the translation activity, organised into three separate lists based on the topics covered. The terms in *List A* were selected from lectures covering the basic fundamentals of programming and Python syntax. Terms in *List B* were chosen from lectures on choice (if statements), loops, and mutability/immutability. Finally, terms in *List C* were taken from lectures on lists, user-defined functions, and modular design.

List A	Literal Value	Source Code	Program	String
List B	Condition	Statement	Expression	Operator
List C	Sequence	Parameter	Index	Method

Table 6.1: The computer terminology used in the translation activity

The activity was repeated three times during the course. The first set of terms (*List A*) was assigned to students for translation after covering all related topics in the lectures. Students were given one week to complete and submit their translations, and the peer review was required to be completed in the following week. The same process was repeated in the subsequent weeks for *Lists B* and *C*. The voting sub-activity was combined for all three lists of terms and completed after all reviews were finished.

The CTT was used as a post-test to explore the impact of the translation activity on students' knowledge of computer terminology. The test was made available to students to complete in the middle of the semester, while the translation activity continued until the end of the semester. This decision was based on the timing of the same test in previous years for the control group. The goal was to make the CTT results more comparable for both groups.

6.4.4 Students' Feedback Survey Design

At the end of the semester, the students in the experimental group were invited to participate in an online feedback survey about the translation activity. The first question was a multiple-choice question with eight statements. The purpose of this question was to explore whether the students agreed with the expected benefits of the activity. For each statement, the students could choose one of five options: '*strongly agree*', '*agree*', '*not sure*', '*disagree*', or '*strongly disagree*'. The eight statements are as follows:

1. *Completing translations of terms improved my understanding of some programming concepts.*
2. *Completing translations of terms improved my understanding of lectures and learning materials (textbooks, quizzes, assignments).*
3. *Reviewing other students' translations was helpful in improving my understanding of some programming concepts.*
4. *Reviewing other students' translations was helpful in improving my next translation task.*
5. *Participating in the translation activity had a positive impact on my learning of programming.*

6. *I would like to participate in another translation activity in my next programming course.*
7. *Knowing there would be voting on the best translation encouraged me to do my best in this activity.*
8. *I enjoyed voting on the best translation, and I was excited to see the voting results.*

The multiple-choice question was followed by three open-ended questions asking the students about what they liked or disliked about the activity, as well as their suggestions to improve it. No personal or demographic information was collected in this survey.

The survey was provided to the students in their native language (Simplified Chinese) to collect more accurate responses and maximise the participation rate. The translation of the survey and the responses to the open-ended questions was done by a professional translator.

6.5 Experiment Results

6.5.1 The CTT Results

The results presented in this section are based on 64 participants from the experimental group and 83 from the control group. The boxplot in Figure 6.2 compares the data collected for the two groups regarding their performance in the CTT, which was distributed at the same period of the semester for each group. By looking at this boxplot, it is easily noticeable that the experimental group performed better in the CTT than the control group.

The boxplot also shows some mild outliers, represented by small circles with case numbers. Mild outliers are values that are more than $1.5 \times$ the interquartile range (IQR) below Q1 or above Q3 [328]. Outliers can only be removed if they are the result of an error in the data; otherwise, they may be valid observations with natural variation [329]. After reviewing the data, it was decided not to remove those values, as they are not expected to be a result of error but reflect the usual diversity in students' performance and are not extreme outliers.

Lastly, the size of the box in the boxplot shows the variance in students' performance. A shorter box means less variance in students' performance on the test [330], as seen in the case of the experimental group.

Moreover, by looking at the detailed descriptive summary in Table 6.2, all indicators show higher numbers in favour of the experimental group. The average score in the

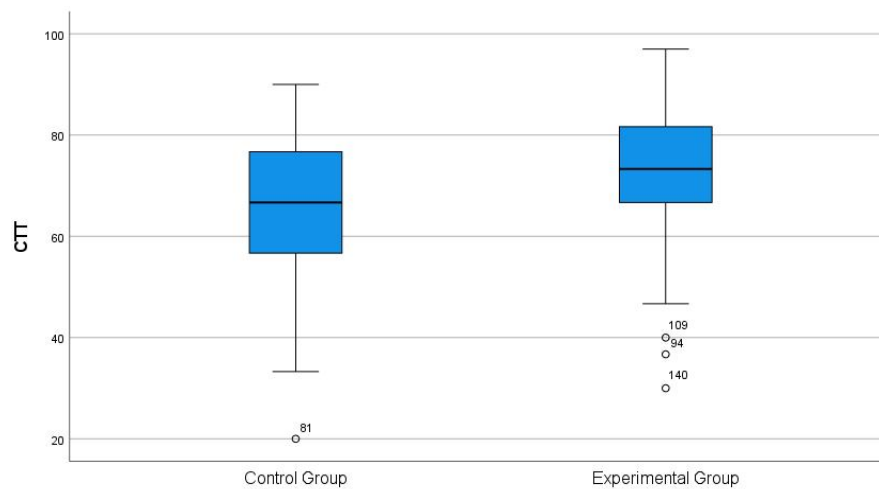


Figure 6.2: Students' performance in CTT

experimental group is higher by approximately 8 points with a lower standard deviation. The median also shows very similar figures to the average, especially in the experimental group. The maximum score is also higher in the experimental group by 7 points, while the minimum shows a 10-point difference. Based on the values of Q1 and Q3, it can be said that half of the students in the control group scored between 56.7 and 76.7, with an IQR of 20 points. In contrast, half of the students in the experimental group scored between 66.7 and 82.47, with an IQR of 15.7, indicating higher performance and less variance among students' performance.

	Control Group <i>n</i> = 83	Experimental Group <i>n</i> = 64
Mdn.	66.7	73.3
μ	65.7	73.12
Std. Deviation	15.14	12.7
Maximum	90	97
Minimum	20	30
Q1	56.7	66.70
Q3	76.7	82.47
IQR	20	15.77

Table 6.2: Descriptive summary comparing the control and experimental groups' performance in CTT

A two-sample t-test was conducted to investigate the significance of the difference in the means of the two groups. The result shows a significant difference in mean CTT score between the control and experimental groups ($t(143.937) = -3.527, p = 0.045$). The

negative t value indicates that the CTT scores of the second group (the experimental group) were higher than those of the first group (the control group). Since all measures were taken to ensure that the two groups were similar, it is highly expected that the difference in their performance is a direct result of the application of the translation activity. Thus, the result can be interpreted as a positive improvement in the students' knowledge of computer terminology as a result of participating in the managed and peer-reviewed translation activity.

6.5.2 Students' Feedback Result

The first question in the feedback survey asked the students if they agreed or disagreed with a number of statements about participation in the translation activity. The options ranged from '*strongly agree*' to '*strongly disagree*' with a neutral option in the middle. For the sake of simplicity, the analysis combines the results of '*agree*' and '*strongly agree*' together, as well as '*disagree*' and '*strongly disagree*' (Figure 6.3). In general, of the 67 responses, over 80% of the surveyed students agreed that the translation activity benefited their learning process in programming. All parts of the activity, including the translation, peer review, and voting, received highly positive feedback from the students.

Looking at the result of each statement, it can be seen that 85% of the students agreed or strongly agreed that the translation activity improved their understanding of some programming concepts, as well as the lectures and learning materials. 84% of the students agreed or strongly agreed that the translation activity had a positive impact on their learning of programming. Reviewing other students' translations was helpful in understanding programming concepts for 82% of the surveyed students. 81% of the students said that this was also helpful in improving the next translation task they had to complete. The same percentage also agreed that the voting part of the translation activity encouraged them to do their best when completing the translation tasks. 76% of the students enjoyed the voting task and were excited to see the voting results. 73% of the students who completed the survey stated that they would like to participate in another translation activity in their next programming course.

The second question in the feedback survey asked the students what they liked about the translation activity. This was an open-ended question to allow the students to express their opinions freely. 28 students answered this question, and six themes emerged from their answers, as shown in Table 6.3. Some of the responses contained more than one idea, so they contributed to more than one theme. The top theme was that the translation activity is generally fair, good, helpful, and exciting, which ap-

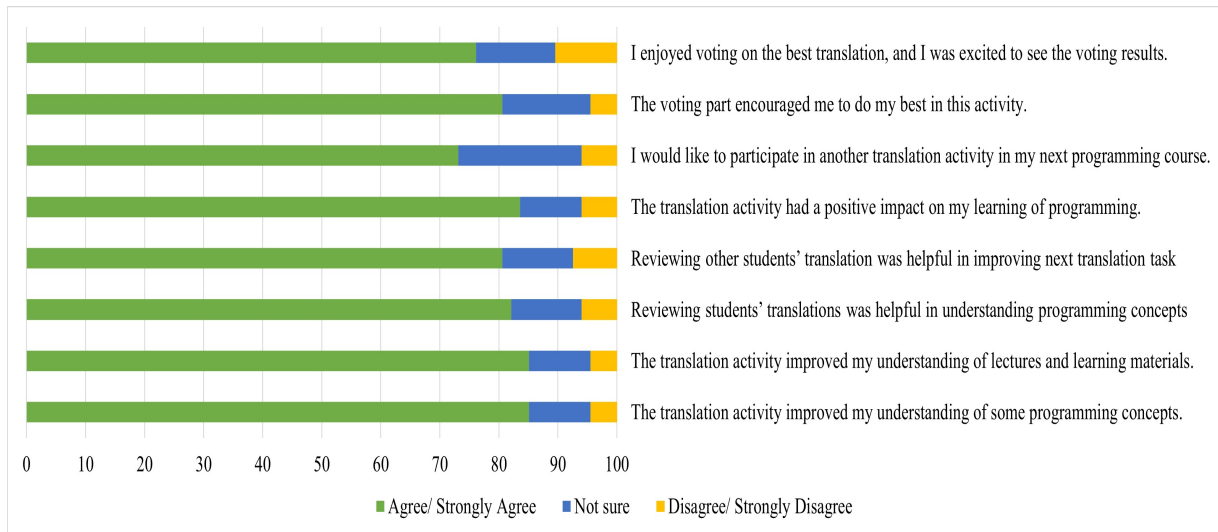


Figure 6.3: Students' feedback on the translation activity

peared in 46% of the answers. 29% of the answers also expressed the benefit of this activity in improving and deepening their understanding of programming terms, concepts, and programming language. 14% of the students found it very interesting and helpful to see other students' translations, which can improve their understanding of some terms and help them evaluate their own translations. 11% of the students also see this activity as an opportunity to increase their knowledge in general. The same number of students also believe that this translation activity is a good way to improve engagement and interactivity in programming classes. One student thinks it also helps improve their understanding of the English language.

Themes	<i>n</i> of responses contributing to the theme (%)
Generally fair/ good/ helpful/ exciting activity	13 (46%)
Help in understanding programming terms/ concepts/ language	8 (29%)
Reviewing other students' translations is very interesting/ helpful	4 (14%)
Increase knowledge generally	3 (11%)
Increase interactivity/ engagement in programming class	3 (11%)
Improve English language learning	1 (4%)

Table 6.3: What students like about the translation activity

It is worth mentioning that the inductive coding method was used to analyse the responses to this question, as well as the next two open-ended questions. The same approach was applied in the analysis of the open-ended questions that appeared in the background survey. Additional information about this analytical approach was presented in Chapter 3: section 3.4.2.

The third question was also an open-ended question that asked the students what they disliked about the translation activity. 27 students responded to this question, and five themes emerged from the analysis of those responses (Table 6.4). The majority of the responses (71%) indicated that there was nothing to dislike or that everything was good. 11% of the respondents claimed that some of the translations submitted by their peers were directly copied from the internet, which could reduce the usefulness of the activity, in their opinion. Seven per cent of the surveyed students said that they faced some technical or connection problems while completing the translation activity. One student mentioned that the process of the translation activity was not easy enough to follow. Lastly, another student said that the translation activity was not helpful.

Themes	<i>n</i> of responses contributing to the theme (%)
Nothing to dislike	20 (71%)
Some translations are copied from the internet	3 (11%)
Technical/ connection issues	2 (7%)
The process of the activity is not easy enough	1 (4%)
The translation activity is not helpful.	1 (4%)

Table 6.4: What students dislike about the translation activity

The last question in the feedback survey asked the students about their suggestions to improve the translation activity. Ten students responded to this question, and the main suggestions are listed here:

- Add more terms to the translation activity,
- Publish the teacher's translation or a standard answer,
- Allow for more answers/add a box for explanations,
- Create a forum for students to discuss the translation of technical terms,
- Link each translation activity with lessons in class,
- Conduct an interview/oral test to prevent cheating,
- Remove peer review and voting and focus on participating in the translation,
- Add activities that include interpreting a line of code.

6.6 Conclusion

This chapter introduced an online learning activity that aims to help students increase their knowledge of computer terminology used in the CS1 class. The motivation for developing this activity was the highly significant correlation that was noticed between the students' scores in the CTT and their academic performance in the CS1 course. Moreover, this chapter showed the results of a quasi-experiment that was conducted

to evaluate the efficacy of this learning activity and its tool. The students' feedback on this activity was also considered and highlighted.

The next chapter discusses the results of the quasi-experiment as well as the English-related factors correlations results that were presented earlier in light of previous research in the area. More focus will be given to answering the research questions and revisiting the research hypotheses based on the presented findings.

DISCUSSION

7.1 Introduction

This chapter delves further into the research findings that were presented in the previous chapter and revisits the research questions and hypotheses presented in the first chapter. It addresses each research question and its related hypothesis, where applicable, separately and discusses the relevant findings in the context of existing research in the field. By connecting the literature review to the findings of this study, this chapter draws meaningful conclusions with regard to the proposed research questions. Finally, a conceptual model that visually summarises the discussed results to illustrate the relationship between the different variables in the study is presented and explained.

7.2 Research Question One and Hypotheses Evaluation

(RQ 1:) What is the influence of the following three factors on NNES students' academic performance in the CS1 course:

In this question, three factors related to the English language and their impact on NNES students' academic performance in CS1 courses were considered. The correlations between the students' scores on different tests, their assignments, final exams, and overall course scores were examined. The following section discusses the main findings for each of the investigated factors, along with the hypothesis related to each of them.

1. English Proficiency Level

The analysis showed a low to medium positive correlation between students' English proficiency levels and their overall academic performance in CS1 courses. The observed correlation was ($r = .211, p < 0.01$) in Case Study One (China) and ($r = .223, p < 0.05$) in Case Study Two (Saudi Arabia) between students' scores in the standard English test and their overall CS1 course scores. This indicates that students with higher English proficiency in the standard test performed better in the CS1 course. This result aligns with many previous studies that have shown a similar level of correlation between students' proficiency in English and their academic performance in programming or other related CS courses [53, 54, 30, 55, 31, 56, 57, 58, 59]. It is also consistent with the results of studies conducted on middle and elementary school students, where their English proficiency was found to correlate with their academic achievements in CS courses [60, 61].

However, the correlation detected in this study is not as strong as that recorded by some researchers [31, 56]. In the study by Ben Idris and Ammar [31], the programming courses were at different levels, and English proficiency was based on self-rating by the participants. Self-rating of English proficiency may lead to inaccurate data, as NNES students could experience low self-confidence in EMI courses [118]. In another study by Wibowo and Tandiono [56], a standard English test was used to measure language proficiency; however, it is unclear which levels of programming courses were included in the study. Despite the differences in the strength of the recorded correlations in these two studies, it is evident that students' English proficiency impacts their academic achievements in CS1 courses.

The impact of English can be influenced by various factors, such as individual English proficiency, the complexity of the language used in class and learning materials, classroom language policies — especially when students and teachers share a language other than English — the teaching approach, and many other elements. All these factors can contribute to varying levels of cognitive load that students may experience in EMI classes. Cognitive load could be a key reason for the differing correlation results between English proficiency and students' academic performance in CS1. Some researchers have suggested that NNES students may experience cognitive overload in EMI programming courses as they struggle to grasp complex programming concepts presented in a language they are not fluent in [39, 10, 77, 78]. This cognitive overload is often associated with lower academic performance; however, this factor was not covered in the current study. Moreover, it has not received sufficient attention in

existing literature, particularly concerning NNES students in programming courses, highlighting the need for further investigation in the future.

One of the interesting findings of this study is the difference in the correlation results of English proficiency levels between the two case studies, particularly in the assignments. In Case Study One (China), assignments, final exams, and overall course scores significantly correlated with English proficiency level. In Case Study Two (Saudi Arabia), the final exam and overall course scores showed a significant correlation, but the assignment scores did not, even though the students in this case study had a lower average English competency than those in Case Study One (China).

The exact reasons for this result are difficult to determine, but the ability of the students in Case Study Two (Saudi Arabia) to communicate with their teachers in class using their native language could be one of the influencing factors. In a related study, students in bilingual classes asked more questions than students in EMI classes [174, 175]. In this study, students in Case Study Two (Saudi Arabia) might have asked more questions about the assignments, which could have reduced the impact of the language barrier on their assignment achievements. However, in the final exam, a strict rule of using only English was applied; therefore, students were not allowed to ask questions in their native language or receive translations from their teachers. This might have increased the impact of their English competency on their exam performance, resulting in a significant correlation that was not observed in the assignment scores. In Case Study One (China), students were unable to communicate with teachers in their native language, either in class or during the final exam session, which may have contributed to the consistent correlation results between the assignment and final exam scores.

Of course, there are a number of studies that conflict with the results of this investigation, finding no significant correlation between English proficiency and academic performance in programming [6, 70, 71, 72, 73, 74, 68]. It is important to note that all of these studies were conducted in countries where English is either the first or second official language, and in most cases, English grades from high school were used as a measure of English competency. Some of these studies also failed to report whether the participants were native or non-native English speakers, meaning their results may not be applicable to NNES students who are still language learners. In this thesis, the two case studies are from non-English-speaking countries, and all the participants are English learners. This may explain why the results differ from those of studies conducted in English-speaking countries.

- Evaluation of Hypothesis 1

H1: *The level of English language proficiency impacts NNES students' academic performance in CS1 courses.*

Null hypothesis: *The level of English language proficiency does not impact NNES students' academic performance in CS1 courses.*

One of the main objectives of this study is to investigate the impact of students' English proficiency levels on academic performance in CS1 courses. Unlike most studies in the area, this study adopted a standard English language test to accurately measure the English competency of the students. The students' scores in the SET and their academic performance in CS1 courses, including assignments, final exams, and overall course scores, were used to test the first hypothesis of the study. The first step in the analysis involved a correlation test to help understand the relationship between the variables. The study included two case studies, each with its unique correlation results, which were particularly different with respect to assignment scores. However, the overall course score showed a similar significant correlation with English proficiency in both case studies, suggesting that the first hypothesis can be accepted.

To obtain more reliable results for hypothesis testing, a multiple linear regression analysis was performed. The results of this analysis did not support the initial correlation findings, as there was no significant impact of English proficiency on the academic performance of NNES students in the CS1 course. The conflict in the correlation and multiple linear regression results indicates that the first hypothesis cannot be definitively accepted or rejected. Further research is needed to determine whether to accept or reject this hypothesis.

2. Knowledge of Computer Terminology

The level of the students' knowledge of computer terminology, as measured by the CTT, correlated significantly and positively with their academic performance in the course ($r = .699$, $r = .813$, $p < 0.01$) for Case Study One (China) and Case Study Two (Saudi Arabia), respectively. The correlation between this factor and students' scores in the CS1 assignment was notably strong and consistent ($r = .566$, $r = .593$, $p < 0.01$) for Case Study One (China) and Case Study Two (Saudi Arabia), respectively. This result is in line with Masrai et al. [33], where they identified a significant positive correlation between the students' scores on a CS-specific vocabulary test and their final course grades in an introductory CS course. These results might explain the negative correlation between the number of difficult computer terminology used in lecture

notes and the students' scores in the programming course, as identified by Keen and Etzkorn [32]. These difficult terms are likely unfamiliar to students, which explains their negative impact on academic performance.

By examining the difference between the correlation results of the two case studies, a significant difference was recorded in the correlation of the final exam with the CTT and, consequently, a significant difference in the correlation of the overall course score with the same factor. No significant difference appeared between the correlation of assignment scores and the CTT, and the actual correlation results for both groups were very close.

Many factors might have contributed to the different results of the correlation between CTT and final exam scores in the two case studies. One of the main factors could be the different cognitive load that the students experienced during the final exam sessions. In both case studies, the final exam included similar topics, but the exam questions were not the same. Each exam was written by a different teacher; as a result, each exam had its unique characteristics regarding the cognitive load it may impose on students. Cognitive load was found to have a significant impact on NNES students' academic performance in the final exam of the database course [41, 21]. This factor is highly dependent on the structure and the language used in formatting the exam questions, as well as individual differences. In their first research, Mason and Seton [41] found that the format and the language used in the final exam paper caused cognitive overload on NNES students, which was the reason for the high failure rate among those students. Using an improved version of the questions based on the guidelines of cognitive load theory (CLT), as well as providing the students with keyword glossaries in the exam reset, helped increase the pass rate. The repeated research [21] confirmed that respecting CLT when writing final exam questions and allowing the students to use glossaries in their native language during the final exam can improve their academic performance significantly. This is also in line with Ehara [98]'s results, where they found that some computer terminology can be confusing for NNES learners and suggested that providing the students with a terminology list can be useful for their learning process.

There is another possible factor that could have contributed to the significant difference between the two case studies in regard to the correlation between the CTT and final exam and overall course scores. This factor could be the number of technical terms used in the lecture notes in the different case studies, which could have impacted the students' academic performance differently. In a study conducted by Keen and Etzkorn [32], they found a significant correlation between the number of buzzwords (words with computer science meaning) and the students' academic performance in the computer science course. In this thesis, the courses in the two case studies covered

very similar topics in a very similar order; however, they were not using the exact same learning materials. The number and level of the technical terminology used in those courses' lectures and materials might have impacted the students' academic performance differently.

Moreover, the CTT was found to explain 49% and 66% of the variability in the overall CS1 course score for case studies one and two, respectively. This is much higher than the CSSVT developed by Masrai et al. [33], which was able to explain only 3.9% of the variance. The main reason for this significant difference in the performance of both tests is the word lists used to build them. The terms used in the CTT were from a special CS1 word list that was generated from learning materials, including textbooks and lecture notes for introductory programming courses similar to the courses included in the study. Moreover, the terms were selected with the help of teachers with experience in teaching CS1 to NNES students, who filtered the most important words in the list.

In contrast, in the CSSVT, the researchers randomly selected a list of vocabulary from CSWL [93], which was generated from conference and journal papers covering 10 different CS sub-disciplines. However, the course in which the CSSVT was applied was an introductory course to CS, covering basic applications used for word processing, spreadsheets, presentations, and databases. Another reason could be the design of the test: for each question in the CTT, the students were provided with a short English definition of the terminology and were asked to select the correct answer from three suggested terms. In the CSSVT, each question showed the students an English CS term, and they were asked to decide if they knew the word or not by selecting 'Yes' or 'No' as an answer.

Finally, no study was found to be in conflict with this finding. Moreover, a number of studies have highlighted the need for students and professionals to master specific field terminology. The result of this study confirms the importance of mastering field terminology from the very early stages of undergraduate study.

- Evaluation of Hypothesis 2

H2: *The NNES students' level of knowledge in English computer terminology impacts their academic performance in CS1 courses.*

Null hypothesis: *The NNES students' level of knowledge in English computer terminology does not impact their academic performance in CS1 courses.*

The second factor investigated in this study was the impact of students' knowledge of English computer terminology on their academic performance in CS1 courses. For this

purpose, a specially designed Computer Terminology Test (CTT) was used as a tool to measure the students' knowledge of computer terminology related to the course.

To test the second hypothesis, the students' scores in the CTT and their academic performance in the CS1 course were analysed. The recorded correlations for this factor were highly significant in both case studies, and the multiple linear regression results supported the correlation findings.

Thus, the second hypothesis was accepted, and the null hypothesis related to it was rejected. The detected correlation is positive, which suggests that students with better knowledge of English computer terminology could achieve higher academic performance in CS1 courses.

3. Level of Foreign Language Classroom Anxiety

The FLCA level was found to correlate negatively with the students' academic performance in programming assignments ($r = -0.193, p < 0.01$) in Case Study One (China). However, no significant correlations were observed between FLCA level and the final exam or overall course score. Similarly, this factor showed no significant correlations with any aspects of academic performance in Case Study Two (Saudi Arabia).

Moreover, most of the courses included in this study were delivered online due to the COVID-19 pandemic, which might have impacted this result. It is possible that the availability of recorded lectures reduced the level of FLCA students might have experienced. This speculation is based on the findings of a study on the impact of online teaching methods on foreign language anxiety in language teaching, where students stated that recorded classes made them feel more confident that they would not miss anything [331].

However, it was expected that the final exam, in the form of a paper-based assessment completed in English, would be more negatively impacted than assignments, where the final deliverable would be primarily code. The analysis instead displayed a higher negative correlation between FLCA level and assignment scores, which was significant in Case Study One (China), even though the average English level of students in this group was higher than that of students in Case Study Two (Saudi Arabia).

The overall course score is determined by a number of different components that are completed and evaluated in different contexts, which might have reduced the influence of FLCA levels on the overall course score. However, regarding the differences between the correlations of the final exam and assignment scores with the FLCA level,

it is difficult to determine the exact reasons for this variation. It could, however, be related to the differences in the nature of these two assessment methods.

Exams typically consist of a number of separate questions, which means that failing to correctly understand one question will not necessarily affect the rest. In contrast, assignments usually involve a single programming problem that contains a series of interconnected components. In order to produce the final code, it is necessary to fully comprehend all of these components. The students' discomfort in asking questions in English might have prevented them from requesting clarification when they faced difficulties in understanding the assignment.

There is a clear explanation for the difference in the degree of significance that the FLCA level has on students' achievements in assignments between the two case studies. In Case Study One (China), the instructors were native English speakers who did not speak the native language of the students, while in Case Study Two (Saudi Arabia), the instructors spoke English as a second language and shared the native language with their students. This allowed students in Case Study Two (Saudi Arabia) to communicate with their teachers in their native language, which was not possible for students in Case Study One (China).

This is highly consistent with the results of Khan and Nabi's research on the use of English versus native language in programming classes [146]. They noted that even students who showed advanced English writing and reading skills still preferred to communicate orally using their native language and were unwilling to engage if they were forced to speak in English.

Similarly, Carter et al. [40] observed that some of their NNES students avoided asking questions in English, which resulted in frequent misunderstandings of assignments and attempts to solve problems that were much more complex than requested. In their research, they explained this behaviour of the students as a lack of comfort with English, fear of being wrong, or showing a lack of understanding, which prevented them from developing programming skills at the same level as their native English-speaking peers.

Discomfort with English can be considered one of the factors related to FLCA, especially when speaking with native speakers, which is one of the factors measured by the FLCAS, the instrument used in this research [242, 249]. In another study, 'embarrassment asking for help' and 'self-doubt' were identified as the main factors for stress among programming NNES students in EMI [145]. These two factors might have contributed to the significant correlation that appeared in Case Study One (China).

The strength of the correlation between two variables can be influenced by various underlying factors related to the nature of the course, the teaching approach, and the students' cultural backgrounds. In this study, the results varied between the two case studies, likely due to differences in the language policies used in the classroom, the students' cultural contexts, and other undetected factors. The same reasons might have contributed to the differences between the results of this study and previous studies.

Overall, the negative impact of FLCA on students' achievements is generally consistent with many previous studies, particularly in the field of language education [332]. However, the significant correlation between FLCA level and assignment score found in Case Study One (China) contrasts with the work of Alghazo and Al-Wadi [333], where they found no significant correlation between this factor and students' mathematics achievements. Notably, most of the students in Alghazo and Al-Wadi's study reported experiencing a medium level of FLCA. Nonetheless, the correlation between the overall CS1 course score and the FLCA level in the two case studies can be considered consistent with what was reported by Alghazo and Al-Wadi [333].

- Evaluation of Hypothesis 3

H3: Foreign language classroom anxiety level impacts NNES students' academic performance in CS1 courses.

Null hypothesis: *Foreign language classroom anxiety level does not impact NNES students' academic performance in CS1 courses.*

This study is the first to investigate the impact of FLCA level on NNES students attending EMI CS1 courses. A well-known scale has been adopted to measure the FLCA levels that NNES students might experience in these courses.

The analysis of the students' scores on the FLCA scale and their academic performance in the CS1 course showed a significant but low correlation, only in one of the case studies and particularly with their assignment scores. Furthermore, the multiple linear regression analysis did not show any significant impact of this factor on the students' academic achievements.

Consequently, the third hypothesis was rejected, and as a result, the null hypothesis had to be accepted. However, it is important to remember that most courses included in this study were provided online, which might have impacted the results. More research is needed to confirm that this result can be applied to traditional CS1 courses.

7.3 Research Question Two

(RQ 2:) To what extent can online managed and peer-reviewed translation activity for computer terminology improve students' knowledge of terminology related to the CS1 course they are attending?

In this activity, students were assigned a number of terms related to their CS1 courses to translate into their native language. The submitted translations were then reviewed by teaching assistants before being distributed to the students for peer review. At the end of the course, the students voted on the best translation for each term and created a glossary for the course based on the voting results. The entire activity was completed through a specially designed online tool to facilitate its implementation.

To evaluate the effectiveness of this activity and its tool, a quasi-experiment was conducted, considering the students who participated in the translation activity as the experimental group. The control group consisted of students who had attended the same course with the same instructors in the previous two academic years. Both groups voluntarily participated in the same computer terminology test at the same time in their course, but the control group had not been offered any terminology-related support.

The group that participated in the managed and peer-reviewed translation activity scored significantly higher on average on the CTT compared to the group that did not. The learning activity, specifically designed to target CS terminology, provides a reasonable explanation for this difference. Thus, it can be concluded that this activity was successful in increasing the students' knowledge of the terminology needed in the CS1 courses. Moreover, according to a previous study in the field, pre-teaching CS vocabulary improved NNES students' achievement in similar courses [76]. Consequently, the implemented managed and peer-reviewed translation activity might improve the students' academic performance in the CS1 courses.

The experimental group was also invited to provide feedback on the activity, and the results support the observed improvement in the CTT results. The majority of students who responded to the feedback survey found the activity very helpful in improving their understanding of the learning materials. In a previous study, teaching CS terminology through computer games was rated by students as very useful for the programming course they were attending [208]. This result also aligns with findings from other studies, which suggest that explanations of terms in the learner's native language increase the comprehension of learning materials [204].

Simanjutak [209] proposed an approach to teaching CS terminology using a mobile app and observed an improvement in the students' knowledge. However, the students in that study highlighted the need for human interaction, which was missing in the proposed learning application. In this study, the peer-review component of the activity appears to have added value in deepening the students' understanding of specific terms and their ability to complete the next translation task. This result aligns with previous studies linking peer-review learning activities with a noticeable improvement in domain-specific skills [313].

Many students also agreed that they enjoyed voting for the best translation, and it was encouraging for them to do their best in this activity. Related studies on the use of electronic voting systems in education have demonstrated their positive impact on student engagement and outcomes [317]. It is possible that the interactivity, in the form of peer review and voting, involved in this activity positively influenced its impact on learning and students' acceptance. Moreover, nearly seven out of ten students expressed interest in participating in a similar activity in their upcoming programming course, emphasising the positive impact this activity has on their overall experience.

Other research has suggested different approaches for improving students' knowledge of CS terminology, as presented in chapter 2 of this research. Most of the proposed solutions require a lot of preparation from teachers and various resources, such as additional teaching staff, smaller class sizes, or the design of specialised learning materials. In many cases, the proposed approach also relies on passive learning, where students receive all the information from the teacher. However, Chambers [212] recommended a collaborative learning approach in programming classes to enable students to learn from each other.

The proposed managed and peer-reviewed translation activity is lightweight, easy to implement, and encourages students to interact with their peers while collaboratively building a glossary for their course in their native language. Research has found that some computer terminology can be confusing for NNES learners and suggested that providing them with a list of such terms can aid their learning [98].

In this study, the glossary that students collaboratively created is intended to help them learn and retain difficult terminology. Future studies could explore whether allowing NNES students to use this glossary during final exams can enhance their academic performance in CS1 courses. A study on database students found that the availability of glossaries in students' native languages was significantly helpful in improving their academic achievement [21].

Furthermore, peer review is believed to help students develop transferable skills such as critical thinking and problem-solving, following and developing evaluation criteria, the ability to fairly and objectively judge other people's work, as well as the ability to assess themselves and benefit from others' reviews to improve their own work [316, 315, 314]. Thus, the proposed activity in this research may offer more valuable benefits than merely increasing students' knowledge of computer terminology.

7.4 Research Question Three

(RQ 3:) What are the perspectives of NNES students about using the English language in CS1 courses?

In this regard, the students' perspectives on four main aspects were explored. First, the use of English as the primary medium of instruction in CS1 courses. Second, the use of English-based programming languages in their first formal programming course. Third, they were asked whether they considered English a barrier to learning programming, given the heavy reliance on English in this field. Finally, their opinions on the most preferred language(s) for learning programming were explored. The following section discusses the main findings.

(RQ 3.1:) What difficulties do NNES students perceive in EMI CS1 courses?

The main difficulties reported by students in relation to the use of EMI include challenges in asking and answering questions, understanding programming concepts, understanding computer terminology, understanding learning materials, and general English language comprehension. This result is highly consistent with the findings reported by Raffaa and Ali [167] who investigated the challenges Saudi students face in EMI CS majors and their attitudes toward it. In their findings, many students reported difficulties in writing, comprehending lectures, and communicating with teachers.

Difficulties in asking and answering questions could make basic communication between the teacher and students extremely challenging. NNES students have been observed to face the issue of misinterpreting programming assignments, which often leads them to attempt solving more difficult problems than requested by the teacher [40]. Additionally, communication problems could be one of the factors contributing to low student engagement in EMI classes, as noted in previous research [173, 172]. This difficulty may not only be related to low English language com-

petency, as it might appear, but could also result from embarrassment in asking for help, discomfort with English, or cultural factors, as recorded in other studies [40, 145].

The students in this study also faced difficulties in understanding programming concepts when explained in English. This aligns with findings from Raffaa and Ali [167]'s study, where some students reported difficulties with English terminology that they believed hindered their understanding of computer science concepts. This difficulty in comprehending programming concepts can also affect native English-speaking students [128], as it requires problem-solving and mathematical skills. However, language skills, including English as a foreign language, were found to be more important than mathematical skills for programming students [62]. This makes complete sense, as if students are not able to fully understand the language used to explain programming concepts, they are not expected to comprehend those concepts correctly, regardless of their mathematical skills.

The issue of NNES students struggling with understanding CS terminology has also been recorded in other studies [10, 94, 92], yet very little work has been done to help them. Similarly, NNES students face many difficulties in understanding programming learning materials in English, which is the only option available to them in most cases [10, 71, 116, 77]. Low English competency and unfamiliar terminology are contributing factors in this situation. Finally, finding learning resources in the students' native language or translations of the CS terminology used was mentioned as challenging by some students, which supports the finding of Khuwaileh [334].

The issues mentioned by students as difficulties in studying programming in EMI classes were consistent with the correlation results of this study. English proficiency and knowledge of computer terminology showed a significant correlation with the students' performance in the course. Based on this, it is possible to understand the recommendation for instructors to consider assessing students' knowledge of the CS subject separately from their English proficiency level [57]. This recommendation is rooted in the understanding that a student's mastery and application of programming concepts may not necessarily align with their mastery of the English language. In contexts where students are learning in a foreign language, their scores on CS assessments may be skewed by challenges arising from language issues rather than a deficiency in the grasp of concepts in the subject being taught. In CS1, this separation in the assessment of programming knowledge and English proficiency would provide a more equitable evaluation of students' programming skills while reducing possible disadvantages to NNES students.

However, there is a need for further research into the most appropriate and fair means of assessing NNES students in CS courses. This should include methods that allow

for the assessment of programming knowledge and skills, or any other technical skills, without being biased by potentially insufficient English proficiency.

(RQ 3.2:) What difficulties do NNES students perceive when studying English-based programming languages?

The two main difficulties reported by NNES students were dealing with error messages and the English interface of the IDE, particularly when using hints and tips. It is widely recognised that error messages are one of the main challenges for novice programmers in general [125]. Some research suggests that students with lower English proficiency might find them more challenging than native English speakers or those fluent in English [123, 124]. In a comparative study, error messages were the most significant challenge for NNES students, whereas for native English-speaking students, they were not a major issue [128, 129]. The findings of this study confirm that error messages are among the top challenges NNES students face when learning English-based programming languages.

The English interface of the coding application used in programming class makes it difficult for some students to benefit from the hints and tips features available in these applications. Some previous research investigated the challenges of using the English interface in programming for NNES students but focused mostly on block-based programming environments [13, 105, 163]. However, all of them agreed that students will either learn faster when using an interface configured to their native language or at least feel that they can do better in programming if they have this option. No doubt that what applies to block-based programming cannot be directly applied to more advanced programming environments. However, one might speculate that this is the same reason why the students in this study find the English interface for IDEs one of the difficulties in learning programming.

Difficulties in intuiting the meaning or functionality of programming language keywords were also faced by some students. Keywords in programming languages that are closer to English, such as Python, were found to be more intuitive for students compared to other programming languages, such as Java [106]. However, the ability to intuitively understand the meaning of English-based programming language keywords might not be the same for NNES students if their general English proficiency is significantly low [107].

In the open-ended questions, some students specified that distinguishing similar programming language keywords was a major problem when writing or reading programming code. Other research that included middle school students concluded that

English-based programming keywords can be difficult for NNES students learning programming [60, 78]. Thus, this study extends the findings of previous research and suggests that both school-age and university-level NNES students may struggle with programming keywords. As suggested by Qian and Lehman [60], students' limited knowledge of keyword meanings and spelling might explain the difficulties they face.

Comprehending programming code was also found to be challenging for some NNES students. Certainly, problem-solving skills and computational thinking are essential for code comprehension, which is most likely not related to the student's native language. However, code comprehension is possibly impacted by the learner's level of English mastery, especially when it is written in an English-based programming language [103]. Factors such as difficulty understanding programming keywords and identifier names in the written code are usually related to English proficiency, while they are also important for code comprehension. Thus, students who are not able to understand keywords or identifier names will consequently face difficulty comprehending programming code.

Finally, remembering the correct spelling of programming keywords, writing code using English letters/alphabet, and reading programming code were recorded as the least difficult tasks in this research. However, it was surprising that, even if only a small number, some students found writing code using English letters/alphabet either difficult or very difficult, particularly in Case Study One (China). It is important to remember that the students participating in this study have either Chinese or Arabic as their native language, both of which have a completely different alphabet and writing system from English. The native language writing system, particularly the use of punctuation, was identified as the main cause of syntax errors for Chinese middle school students [37]. This does not exactly explain the challenges some students faced in using English letters/alphabet in this study, but it highlights the possible impact of the students' native language on their learning of English-based programming languages. Another explanation for this might be the poor level of basic English language skills that some students have.

(RQ 3.3:) To what extent do NNES students consider the English language a barrier to learning programming?

Only a small number of the surveyed students consider English an extreme barrier to learning programming. However, only 16% of the students in Case Study One (China) and 30% in Case Study Two (Saudi Arabia) find it not a barrier at all. Consequently, the majority of the students in both groups (65–80%) believe that the barrier presented

by the use of English is between low and moderate. It is difficult to clearly identify the factors that led to this result; thus, the discussion will be mainly speculative.

It is possible that the low English language mastery of the students, as well as the difficulty in understanding CS terminology, might be contributing factors. Another possible factor is the time NNES students might need to study for this course outside the class. A study by Guzman et al. [118] found that NNES students spend significantly more time studying outside the programming class compared to native English speakers. In this research, one Saudi student in Case Study Two (Saudi Arabia) remarked that the main challenge they face is the long amount of time they spend looking for translations when studying. This has also been mentioned by Saudi students in a study by Alhamami [311] as one of the challenges they face in EMI courses. This extra time requirement could be one of the factors influencing students' opinions about the extent to which English is a barrier to their learning. This might be even more influential if the students are enrolled in other courses while studying programming.

A related question asked students to what extent they considered the use of EMI to make studying programming easier or harder. Most of the students believe that using EMI in CS1 courses makes it more difficult. Ruby and Krsmanovic [52] stated that using EMI in programming courses could add another layer of difficulty for NNES students, which aligns with these results.

The interesting difference that appeared between the two case studies is that the group with higher average English proficiency found that EMI makes the course more difficult, and vice versa. In the results, the percentage of Saudi students who think that EMI makes the course more difficult is high (53%), but lower than that of Chinese students (73%). However, the average English level based on the same standard English test was higher among the Chinese students ($\mu = 55.2$) compared to Saudi students ($\mu = 42.5$). The fact that Chinese students report English as more difficult and more of a barrier would seem odd; as such, there must be some underlying factors influencing this.

The two obvious influencing factors are the differences in culture and learning environment. Cultural differences are likely to be difficult or impossible to fully identify. However, some Chinese cultural concepts, such as a feeling of shame or losing face, appeared in the literature as having a significant impact on the Chinese students' willingness to communicate in English [335]. Still, similar factors, including shyness and fear of making mistakes, even though they have not been directly linked to Saudi culture, were found to be impacting Saudi students' performance in English speaking skills [336]. Therefore, the extent to which different cultures might influence NNES

students' opinions about the difficulties EMI might add to CS1 courses is hard to determine.

One major environmental factor would be the native language of the instructors. For Chinese students, the instructors are native English speakers, and they do not speak the native language of the students. On the other hand, the instructors of the Saudi students speak English as a second language and share the same native language with their students. Even though the courses in Case Study Two (Saudi Arabia) mainly apply EMI in all aspects of the courses, students might have had the ability to use their native language to communicate with their teachers in a way that is not possible for students in Case Study One (China).

(RQ 3.4:) What language(s) do students prefer to use in CS1 class?

The majority of the participating students expressed their desire to use both English and their native language in the CS1 course. This result supports the finding of Pal [92], where the students favoured continuing to use EMI and receiving explanations of new concepts or terminology in their native language. In another study, NNES students preferred oral communication in class to be in their native language but written communication to be in English [146]. Moreover, the majority of Vietnamese students in a technology engineering course expressed their desire for a bilingual education system [96]. However, the finding of this research does not match the result of Alhamami [57], where students in the first-year CS program at a Saudi university preferred to study in their native language.

Some studies compared the use of EMI and bilingual instruction using English and the students' native language in some CS courses. No difference has been noticed between the students' academic performance in both settings; however, students were generally more satisfied with being able to use their native language in class [171, 172, 174, 175]. The students in bilingual classes engaged more with the lecture and asked more questions, which were of higher complexity than students in EMI classes [174, 175]. This might have a connection with different factors, including the level of comfort with English and the level of FLCA that some students might experience in EMI courses. NNES students can also experience feelings of embarrassment when asking for help in programming courses, which contributes to increased stress levels [145]. Thus, it is possible that students desire to be able to use their native language in order to reduce the stress caused by being forced to use only English in class.

7.5 Conceptual Model for the Research Findings

The primary goal of developing the conceptual model illustrated in Figure 7.1 is to enhance understanding of the intricate interplay among the studied factors that influence the academic performance of NNES students in CS1 courses. This model systematically organises and visualises these relationships, thereby enabling a clearer explanation of the findings observed in this study. By illustrating the connections between different influencing factors, the model not only clarifies these complex dynamics but also provides a robust framework for guiding future research in this area. This foundational understanding can inform the development of more effective educational strategies to support NNES students in their academic journeys.

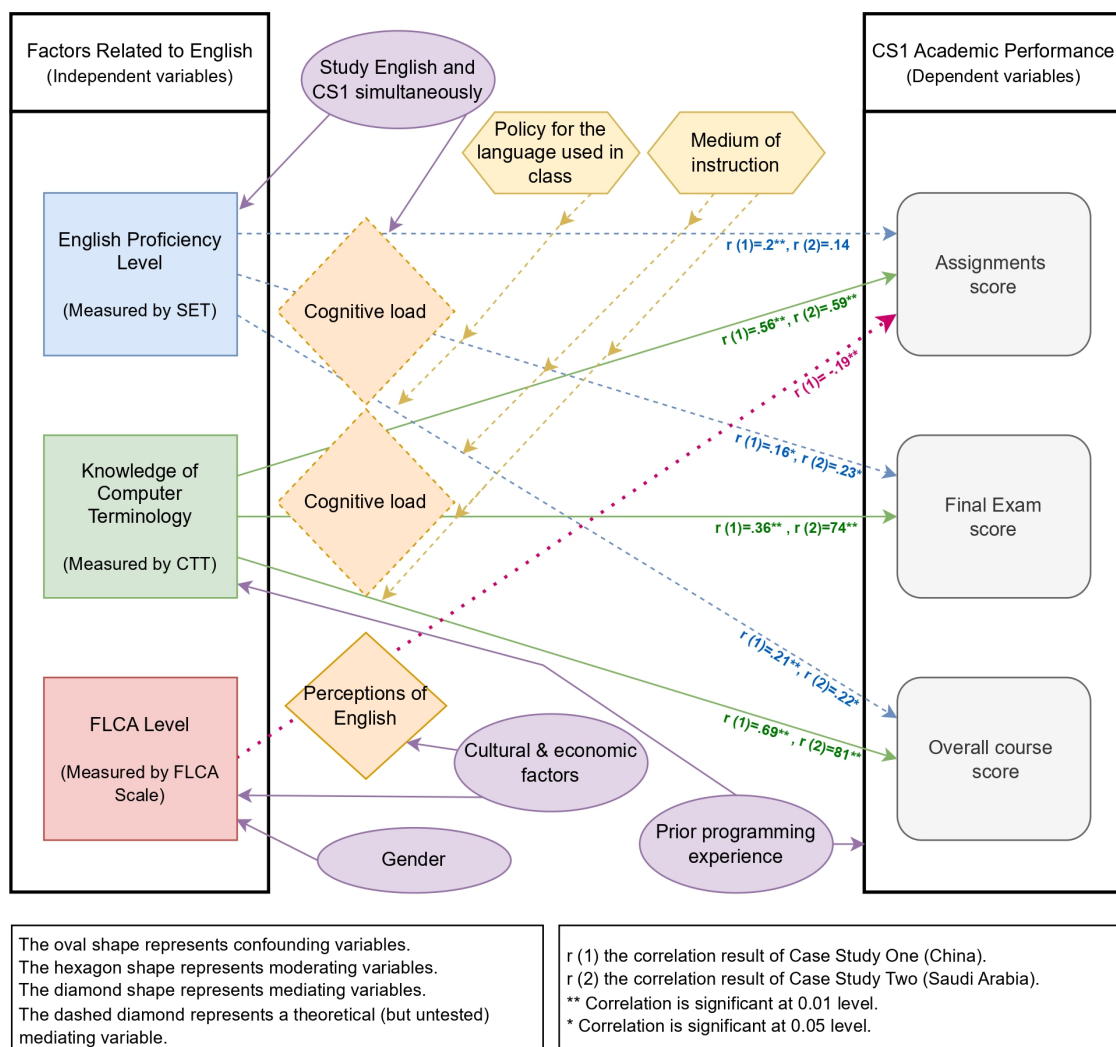


Figure 7.1: Conceptual model

The conceptual model in Figure 7.1 includes different types of variables covered in this study. On the left side are the independent variables, including English proficiency level as measured by the SET, computer terminology knowledge as measured by the CTT, and the level of FLCA measured by the FLCA scale. On the right side of the diagram are the dependent variables that were used to measure the students' academic performance in the CS1 course. These include the assignment, final exam, and overall course scores. The arrows that connect the independent variables to the dependent variables represent the correlation found between them. The model also shows moderating, mediating, and confounding variables that influence the independent and dependent variables or the relationship between them.

By looking at the first independent variable, it is clear that the English proficiency level of the student positively correlates with their academic performance in CS1 across all three measured aspects: assignment, final exam, and overall course scores. The correlation is medium to low but still significant for all case studies. This means that students with higher English proficiency levels are likely to achieve better academic performance in the EMI CS1 course. However, this relationship is not straightforward, as the cognitive load students experience in class might play a mediating role in this relationship. For instance, according to cognitive load theory, if a student has a low level of English proficiency, this will increase the cognitive load they experience when trying to understand complicated programming concepts explained in English. Therefore, a low level of English proficiency can increase cognitive overload, which leads to lower academic performance in the EMI CS1 course. It is important to clarify that cognitive load was not measured in this study, and its explained impact is theoretical.

Of course, the medium of instruction used in the course can be considered a moderating variable because it plays an important role in the strength of the correlation between the students' English proficiency level and their academic performance in the CS1 course. This includes the language of the course, the language used to communicate in class, and the language of learning materials and assessment. In this study, the medium of instruction for all included courses was English. However, the level of the language used in these courses might vary, and as a result, its impact on the correlation level might vary as well. The other important moderating variable that might have impacted the correlation level between English proficiency and academic performance in CS1 is the policy of the language used in the class. For example, in Case Study Two (Saudi Arabia), the teacher shared the same native language with their students, not English, and they were able to communicate using their native language sometimes, as they stated in the survey. They might have been asking more questions about the assignments, which can explain the absence of a significant correlation between the

English proficiency level and assignment score in Case Study Two (Saudi Arabia). The students in Case Study One (China) did not share the same language with their teacher, and therefore they were using English only in all types of communication. Unlike the assignment score, the final exam score showed a significant correlation with English proficiency in both case studies. This can be explained by the strict policy of using only English in final exams in both case studies. In the final exam score, Case Study Two (Saudi Arabia) showed a higher correlation with English proficiency than Case Study One (China), and this can be attributed to their lower average English proficiency level and the fact that they were forced to use only English. The influence of these two moderating variables, the medium of instruction and the policy of the language used in class, can also be extended to the correlation between the English proficiency level and the overall course score, as it is a result of how the students did in assignments and final exams.

There is also a confounding variable that can be linked to both English proficiency and cognitive load. The students in both case studies were enrolled in an English language course in the same semester as the CS1 course. That means their English proficiency level and the cognitive load it caused could have been changing throughout the study period. Studying the English language is supposed to improve the students' proficiency in the language, but the level of improvement depends on different factors, such as individual abilities, course content, teaching approach, and the number of course hours. Additionally, studying English simultaneously with programming has been mentioned in some studies as a contributing factor to increased cognitive overload. In future studies, controlling this factor may enhance the understanding of how English proficiency and cognitive load affect students' academic performance in CS1.

Considering the second independent variable, knowledge of computer terminology, it is possible to see a strong correlation that connects it to all of the aspects used to measure students' academic performance in the CS1 course. The students' prior experience can influence their level of knowledge of such terminology as well as their academic performance in the course in general. In this study, the majority of the students in both case studies had no or very little experience in programming. Therefore, this confounding variable may have little or no influence on the results of this study. However, researchers who intend to investigate this factor in the future are strongly recommended to consider the prior programming experience of the participants, as people with advanced experience in programming are more likely to have a high level of knowledge of computer terminology.

Moreover, the medium of education can be considered a moderating variable in the relationship between the knowledge of computer terminology and the students' aca-

ademic performance in the CS1 course. As explained above, all courses included in this study apply English as a medium of education; however, the level of the language used, the number of computer terms used in each lecture or assignment, and the level of detail provided when the terminology is first introduced to the students can all vary based on the course material. Therefore, the medium of education can play an important role in the level of correlation between the knowledge of computer terminology and the students' academic performance in the CS1 course. Unlike English proficiency, it is not expected that the language policy in the classroom plays a role in the correlation between knowledge of computer terminology and students' academic performance. The reason is that most computer terms will be equally difficult to understand when they are translated into the students' native language, and in most cases, there is even no direct translation for such terms.

Theoretically, the cognitive load might also act as a mediating variable for the correlation between the knowledge of computer terminology and students' academic performance in the CS1 course. Considering that this terminology is all in English and introduced in English to students who are not fluent in the language, it is possible to say that cognitive load theory can be applied here as well.

The last independent variable in this study is the FLCA level, as measured by the FLCA scale. This variable showed only one significant correlation with students' scores in the CS1 assignment in Case Study One (China). An important moderating factor that might have influenced this relationship is the policy of the language used in class and the difference in native language between the teacher and students. In Case Study One (China), a strict role of using only the English language was applied in all communication between teachers and students. This role was less strict in Case Study Two (Saudi Arabia), as the students were able to communicate with their teacher in their native language, which might be the main reason for the absence of a correlation between FLCA level and academic performance in the CS1 course. The impact of this moderating factor on the relationship between FLCA and academic performance is not presented in the conceptual model to avoid complications, as the correlation was not strong.

Moreover, the teachers in Case Study One (China) are native English speakers, and according to FLCA theory, speaking with a native speaker of the language can contribute to an increased level of FLCA. However, the students in this case showed a highly positive perception of using English in the CS1 course, which might have helped reduce the level of FLCA they experienced and, as a result, lower the negative impact that it had on their academic performance. Therefore, the students' perception of using English in CS1 courses can be considered a mediating factor. The differences in the perception of using English between the two case studies can be attributed to the unique

cultural and economic factors that each case has. The difference in cultural background between the two case studies could have also affected the level of FLCA that different groups of students experienced and, consequently, its impact on their academic performance in the course. An additional important factor that was not included in the diagram but is anticipated to influence the FLCA level is the online delivery of most of the CS1 courses in this study. Future research should explore how the teaching method affects this factor and its relationship with the other variables.

Furthermore, while gender has been identified as a factor that influences FLCA levels, its effect on academic performance in the CS1 course and its interaction with other independent and dependent variables were beyond the scope of this study. More research is needed to understand the complex relationship between gender, FLCA, and programming achievement.

CONCLUSION

8.1 Research Overview

The principal aim of this research was to investigate the impact of English on NNES students' academic performance in CS1 courses. An extensive review of the relevant literature subsequently identified three English-related factors for investigation: general English proficiency level, knowledge of computer terminology, and the level of foreign language classroom anxiety (FLCA) experienced by the students. To achieve this aim, a multiple case study methodology and various research instruments were adopted to gain the depth needed to understand the topic.

Three hypotheses were proposed, suggesting that all three English-related factors mentioned above impact NNES students' academic performance in CS1 courses. However, based on the analysis of data collected from the two case studies, a definitive conclusion regarding the first hypothesis could not be reached, meaning that the impact of the students' English proficiency level on their academic performance in CS1 remains unclear. On the other hand, the second hypothesis was accepted, indicating that NNES students' knowledge of computer terminology affects their academic performance in CS1 courses. The third hypothesis suggested that the level of FLCA could impact students' academic performance in the EMI CS1 course, but the results did not support this hypothesis. Nevertheless, the findings of this study do not entirely dismiss the influence of English proficiency and FLCA levels on NNES students' academic performance, as significant correlations were observed with various assessment aspects of these courses. This research demonstrated how English can, both directly and indirectly, influence NNES students' academic performance in such courses. The multiple case study findings were used to develop a conceptual model illustrating the complex relationships between the studied factors and related variables.

As a consequence of the earlier findings, a further aim was adopted to implement and evaluate an online learning activity to help these students improve their knowledge of computer terminology related to the course. A managed and peer-reviewed learning activity was designed, along with an online tool to facilitate the application of this activity. Based on the results of the quasi-experiment, students who participated in this activity showed a significant improvement in their knowledge of computer terminology related to the course, as measured by the CTT. Moreover, the students' feedback on the activity and the tool was highly positive, and the improvements they suggested were presented for future consideration.

Finally, the study explored NNES students' perspectives on different aspects of the CS1 course that are related to English. The first aspect was the challenges related to the use of EMI in these courses. Students reported a number of challenges, the main ones being asking and answering questions in English, understanding programming concepts explained in English, and understanding computer terminology. In regard to the challenges related to using English-based programming languages, the top two challenges were dealing with error messages and the English interface of the coding software. The study also asked NNES students if they think that English represents a barrier to learning programming, and the results showed that the majority of them believe that English represents a low to moderate barrier to their learning. The last question this study asked was which language(s) NNES students would prefer to use in CS1 courses, and the majority chose a combination of English and their native language.

8.2 Significance and Contributions

The main significance of this research lies in the fact that little is known about the impact of English on the academic performance of NNES students in CS1 courses. Focusing on this important research area is a critical contribution to the field of computer science education and a cornerstone for future research in the area. Although the mention of the challenge or impact of English on CS students can be traced back to the 1990s, the number of published studies focusing on NNES students is still small. The number of potentially affected students may have sharply increased with the wide application of EMI in higher education institutions around the globe, which adds extra importance to research on this topic. The identification of clear research questions and objectives that serve to best understand this topic from different perspectives represents an important step towards helping those students and providing them with a more equal learning experience to their native English-speaking peers.

Therefore, this study is the first in-depth investigation focusing on NNES students and the impact of English on their academic performance in CS1 courses. The effect of English on programming students has been mentioned in several studies; however, none of this research has been designed to study this topic in CS1 courses, particularly focusing on NNES students. Moreover, much of this research has been conducted in English-speaking countries (mainly the US) or countries that have English as one of the official languages, which has a significant influence on the number of NNES students taking part in such studies as well as their proficiency level in English. In this research, two case studies from non-English-speaking countries where EMI is used in CS1 courses were included.

The other significance of this research is represented in the research output that can help researchers, educational institutions, educators, and students. An extensive literature review was conducted as part of this study to explore all the work related to the challenges faced by NNES students in programming courses and some other CS-related subjects. In this review, all possible factors that could have an impact on NNES students were discussed, not only the three factors under investigation, which makes this review a great asset to research in this area. Moreover, the studies that compared native and non-native English-speaking students were also reviewed, even though the focus was on NNES students, to give a better understanding of the topic. It is widely acknowledged that the literature review is a time-consuming process; thus, the availability of a state-of-the-art review that covers all aspects of the topic might play a key role in boosting the research work in the area. This could help increase the number of studies on this topic while shortening the research time.

Moreover, the methodology adopted in this thesis, which has been designed to overcome the limitations presented in previous studies, might inspire future researchers. This study is the first to adopt a multiple case study methodology to investigate the difficulties that NNES students face in computer science majors and particularly in CS1 courses. This methodology has a lot of potential and can help understand the situations of students from different backgrounds by focusing on two or more cases. The case study methodology is known to be the best for exploring research phenomena that are not clearly understood and for testing hypotheses. Using a multiple case study approach, rather than a single case study, helps to address the limited generalisability often associated with case study research. The adoption of this methodology in this study not only provides valuable insights but may also serve as a practical example for future researchers, demonstrating how to effectively implement and utilise this approach in their own work.

One of the main limitations in many previous studies in the area is the absence of an accurate method for measuring the English proficiency of the participants. Many studies used English grades from high school records or simply described the participants as non-native, not fluent, or English learners when referring to participants' English competency. Not adopting standard English tests in those studies might be due to the high cost and limited availability of test centres. This research overcomes this limitation by adopting a free online standard test that has a high level of accuracy. The test was originally developed and used by a well-known educational institution specialising in teaching English as a foreign language in many countries. The adoption of this test in this research can inspire future researchers to either adopt the same test or search for other available options.

In addition, this study includes the first re-examination of the influence of the English computer terminology factor on academic performance in programming courses since it was mentioned by Keen and Etzkorn [32]. A novel computer terminology test was developed and validated to achieve this aim. The test can be used by CS1 course instructors to evaluate students' knowledge of course-related terminology. The CS1 word list that has been generated to build this test is also available and can help instructors set learning goals for the course; it can also be provided to students as additional learning materials. It can also be used to shape and focus the language courses provided for NNES students in CS majors. Additionally, teachers and researchers can produce valuable learning materials for advanced programming courses or any other subject by following the steps described in this research for generating the CS1 word list and building the computer terminology test.

This thesis also presents the design, implementation, and evaluation of a new online learning activity to increase students' knowledge of CS terminology related to the CS1 course. The evaluation of this activity showed a positive influence on NNES students' knowledge of the course-related terminology. However, this activity is not limited to programming courses. In fact, this activity and its tool can be used in any other courses and majors where students need to learn field-specific terminology, such as computer science, engineering, and medicine. This activity and its tool can be easily adopted by instructors or course designers to achieve important learning goals. Moreover, suggestions for possible improvements that can be added to the activity and the tool, based on students' feedback, were provided for future researchers to consider if they decide to use this activity in their research.

This study is unique as it first explores foreign language classroom anxiety (FLCA) level as a factor that may affect NNES students in EMI programming classes. It is also the first study to adopt a well-known FLCA scale to measure this factor in an EMI CS1

course. The original FLCAS was designed to be used in foreign language classes [142], however, based on the previous adoption of this scale in different courses by other researchers, an adjusted scale was produced and used in this study. The adjusted FLCA scale provided in this study can help future researchers easily adopt it in their studies. The used scale was adjusted and translated into two languages which can save a lot of time for future research.

Finally, this study is the first to explore NNES students' perspectives on the use of English in CS1 courses. The outcome of this investigation can assist educational institutions in gaining a clearer understanding of the realities within such classes and open more doors for future research. This understanding can help programme and curriculum designers improve the courses provided to NNES students, which may reduce the dropout rate for these courses and create a more equitable learning experience for all students.

8.3 Implications and Recommendations

The study's conclusions provide valuable insight for educational institutions looking to enhance the CS1 course experience for their NNES students. The fact that many students believe it would be helpful to have both English and their native language used in these classes is an important finding. It raises questions about the strict use of EMI in programming classes, especially when both the instructors' and the students' native languages are the same. Allowing some flexibility in language use, whether through bilingual discussions, additional materials, or occasional explanations in students' native languages, could improve comprehension and reduce cognitive load. Obviously, this is not possible for courses where instructors do not speak the native language of the students or in classes with diverse language backgrounds. However, institutions can still implement measures to support students more effectively. Assigning teaching assistants who share the students' native language can be beneficial, as can facilitating group discussions among students who speak the same first language. Implementing a mentoring programme where senior students provide guidance to CS1 students in their native language may also be helpful.

This study also highlights the importance of improving language support, particularly in spoken English skills. Many students reported difficulties in asking and answering questions in English, which affected their ability to engage in class discussions and seek help when needed. English courses designed for EMI students should place greater emphasis on verbal communication, helping students develop confidence in speaking.

Additionally, a key factor influencing students' academic performance in CS1 is their familiarity with computer science terminology. Universities should consider incorporating domain-specific terminology training into language courses to bridge this gap and improve students' comprehension of course materials. One effective approach to address this challenge is to enhance the emphasis on terminology learning within the objectives of CS1 courses. This can be accomplished by incorporating additional supportive learning materials that specifically target the subject terminology and introducing more class activities designed to help students understand the meanings and applications of these terms.

In CS1 classes, instructors play a key role in helping NNES students. Many teachers might not be completely aware of how language obstacles impact their students' educational experiences. Educational institutions should provide proper training for instructors and offer guidance on teaching strategies in EMI, particularly for classes with diverse language backgrounds. Many NNES students find it difficult to take notes, listen to lectures, and understand new concepts all at once. Simple changes, such as providing lecture slides with more thorough explanations, might help students follow along without worrying about forgetting to take notes. Making lecture recordings available could also be beneficial for NNES students, as it allows them to revisit the material at their own pace. Instructors should also carefully phrase test questions, ensuring that the language used in assessments is not unnecessarily complex to reduce the cognitive load NNES students might experience. Moreover, the findings indicated that although FLCA did not show a statistically significant correlation in all cases, it still influenced some students' experiences. Institutions and instructors should be mindful of this and strive to create a supportive classroom atmosphere where students feel comfortable participating.

Assessments should also be designed to ensure that English proficiency does not unfairly impact students' grades. Where possible, instructors should separate programming skills from language skills when evaluating the performance of NNES students. Assessments may be more equitable if students are allowed to explain code orally in their native language or if grading standards are adjusted to prevent small grammatical and spelling mistakes from deducting points. This approach ensures that students are evaluated primarily on their programming ability rather than their English proficiency.

At the policy level, universities should adopt a more flexible approach when applying EMI for NNES students. While the use of English is the core concept of EMI education, strict policies that do not allow students to use their native languages entirely may not be beneficial for their learning process. More flexible policies that allow nat-

ive language support where appropriate should be considered by institutions. Native language use could be helpful if allowed in formative assessments, study groups, support sessions, or one-on-one communication with teaching staff. Moreover, specially designed English training programmes, such as English for specific purposes courses, should be made more widely available to help NNES students bridge the gap between general language skills and subject knowledge.

In summary, small but meaningful changes, such as offering bilingual support, improving English courses, emphasising terminology learning in CS1 courses, refining assessments, and training teachers on dealing with linguistically diverse CS1 classes, can significantly enhance NNES students' experience in CS1 courses. Universities that adapt their policies and teaching methods to better support these students will help create a more inclusive and effective learning environment.

8.4 Limitations and Future Work

8.4.1 Theoretical and Empirical Considerations

The impact of English on NNES students' academic performance in CS1 courses is a topic that used to be surrounded by ambiguity, as very little work has been done to investigate it. Thus, this study had to focus on understanding this issue as a priority instead of seeking generalisable results. However, the inclusion of two case studies rather than one makes the output of this thesis more generalisable. For this reason, large groups of students from two different language backgrounds were included. Yet, it is not clear if the results can be generalised to students from other language backgrounds. The students in the two selected case studies speak either Chinese or Arabic as their native language, which is believed to represent a large population, considering the number of people who speak those languages. However, the alphabets and writing systems of those native languages are completely different from English. Thus, the results of this research might not be directly applicable to students whose native language uses the Roman alphabet, such as some European languages. Future research might consider comparing the influence of English in CS1 courses on students whose native language uses the Roman alphabet with those whose native language uses a different type of alphabet. Additionally, more focus might need to be placed on minority populations, as they might face more challenges, particularly in finding translated learning materials and coding software that support their native languages.

The participants in this study have been accepted to enrol in these courses with some basic English requirements, which means they have at least a basic knowledge of English. Thus, the results presented in this study cannot be generalised to NNES students who have no basic knowledge of English. Future research might consider investigating the impact of teaching English-based programming languages on students' learning when they have no basic knowledge of English.

This study focuses on NNES students, which explains why the selected sample consists solely of NNES students. However, some parts of the research might have benefited from including native English speakers to ensure the generalisability of the results. For example, the CTT was mainly designed for NNES students, but there is no reason why it should not be suitable for NES students as well. However, it was only tested and evaluated in courses including NNES students. Teachers who would like to use this test in classes with NES students are recommended to watch for any possible influence of the students' native language on their performance in this test. Moreover, research is needed to understand the impact of computer terminology knowledge on the academic performance of students who speak English as a native language in programming courses.

During the design of this study, all selected courses were planned to be traditional face-to-face classes. However, as a consequence of the COVID-19 pandemic, all courses were unfortunately converted to online formats. This continued for almost all courses in the three academic years covered in this study. Only the courses in case study two during the second year of data collection were partially online, meaning that some classes of those courses were face-to-face. However, all final exams for both case studies were conducted as paper-based exams throughout the entire period of data collection. The impact of the teaching method is not clear, and the results cannot be generalised to traditional classes. More research is needed to confirm whether the results presented in this study can be applied to traditional CS1 courses.

In addition to converting to online teaching, travelling to different countries to collect data when COVID-19 restrictions were in place was almost impossible. Therefore, the data collection process was arranged to be conducted online at an early stage of the study, as it was unclear when the restrictions would be eased. The influence of using online means to collect the data used in this study is unknown. However, the online survey has been a trusted method used by most researchers for many years, even before the pandemic. Consequently, there is no reason to believe that the online collection method for the FLCAS, background survey, and the students' feedback on the translation activity had an impact on the quality or validity of the data. The English proficiency test is originally an online test; thus, its validity would not change by its

adoption in this study. Regarding the computer terminology test, there is no reason to believe that its results could be impacted by the delivery method, but it would be useful if future research could prove that by comparing the results of online and paper-based delivery for the same test.

In the evaluation of the managed and peer-reviewed translation activity, the results of the experimental group showed a significant improvement in their knowledge of computer terminology. This might result in an improvement in the students' academic performance in the course, as can be concluded from the highly significant correlation result. However, this study did not explore the impact of this learning activity on the students' academic performance in the CS1 course. Future research might consider seeking empirical evidence to prove the possible positive impact of this learning activity on the students' academic performance in such courses.

This study focused on English-related factors impacting NNES students' academic performance in CS1 courses; however, no single study can cover all possible factors. Three important factors were investigated in this study, but research is needed to explore other influencing factors. The impact of cognitive overload on NNES students has been investigated in different CS courses; yet, CS1 is not one of them. Researchers are strongly advised to consider looking into this factor in their research.

8.4.2 Methodological Considerations

The methodology adopted in this research is believed to be the most suitable for investigating this topic, but it comes with its own limitations. When exploring topics with little previous work or conflicting opinions, such as the topic of this research, a case study is the best approach; yet, it is difficult to generalise its results. To overcome this limitation, a multiple case study with two cases was adopted, but another challenge emerged from this decision. Finding the exact same CS1 courses provided in two different countries and institutions is impossible. The courses can differ in content, exams, assignments, grading systems, and the instructors who teach those courses. However, the selected courses were verified to be very similar in content and topics covered, and the grading systems of both institutions are very similar, but with slightly different marking schemes. The students' grades in both case studies were normalised in the process of data preparation to make the results more comparable. There is a possible impact of these differences on the results, but it cannot be easily distinguished. The only major difference that could have impacted the results is the ability of the students in Case Study Two (Saudi Arabia) to communicate with their teachers in their native language. However, the results of each case study were presented separately, and

this difference was highlighted and discussed in the results. In fact, these differences might make the common results between the two case studies more generalisable and provide some explanations for uncommon results.

In terms of the instruments used, to measure general English proficiency more accurately, this study adopted an online standard English test provided by an international education agency. While this test has high accuracy, comparable to other widely recognised tests, it depends mainly on the assessment of reading and listening skills. At the time of conducting this research, there were no available standard tests that measured speaking and writing skills which were affordable and easy to adopt. In the future, researchers are recommended to search for standard English proficiency tests that measure all four skills and might be available.

The second instrument used in this study was the computer terminology test, developed and evaluated thoroughly using statistical methods and data collected from both case studies. The test aims to measure the level of students' knowledge of computer terminology related to CS1 courses. The validity analysis results showed that the test is generally valid and measures what it is supposed to measure. The analysis also highlighted some points that might need improvement; however, no changes were made to the test in the second round of data collection. The decision to continue using the same version of the test was made to ensure the consistency of the data throughout the study, considering that the identified weaknesses had no major impact on the test's validity. Moreover, none of the suggested changes guarantees an enhancement of the test's validity, and the time frame for this study could not accommodate another round of data collection and validity checks, which would be required if any updates to the test were made. Consequently, researchers planning to update this test before adopting it in their research are recommended to allocate time for validity analysis of the updated version. The recommended changes were presented and discussed in chapter 4.

The third research instrument used is an adjusted version of the FLCAS developed by Horwitz et al. [142], which has been used by several researchers in different contexts. The changes made to the original scale were very minor and represent no threat to its validity, as similar changes have been made by other researchers who adopted this scale in their research. As the participants in this study are all NNES students, the adjusted scale was translated into their native languages. The results of Cronbach's Alpha, as calculated in this study, did not show any threat to the instrument's validity that could have been caused by the adjustment or translation. However, future researchers who would like to adopt the adjusted scale are recommended to report Cronbach's Alpha in their results to demonstrate its validity.

In the managed and peer-reviewed translation activity, students were asked to select one of the predefined evaluation options during the peer-review process. However, students were encouraged to provide more detailed feedback in the provided text box, though this was not mandatory. The summative nature of the peer review (i.e. the predetermined selection of options) might reduce the benefits of this activity. Researchers who plan to implement this activity in their class might consider making the formative feedback mandatory instead of optional to increase the learning value of the activity. Moreover, the researcher did not expect the students to translate the terms without looking for meanings or translations that might be available on the internet when participating in the translation activity. However, the students were encouraged to come up with their own translation and not copy it from other sources, but it seems that some of them did not follow this advice. This might have negatively impacted the learning benefits of this activity and demotivated some students. Many factors could have contributed to this, such as the lack of a methodical approach to guide students in writing their own translations or a lack of motivation. In future studies, providing more detailed instructions, offering examples of appropriate translation methods, or conducting discussions on translation choices may help encourage students to engage with the exercise more critically and effectively.

8.4.3 Ethical Considerations

In line with the policy of the granted ethical exemptions, the collected data cannot be shared or made available for other researchers to benefit from. All data collected should be destroyed after the completion of this thesis. This limits the possibility of repeating the analysis or performing further analysis of the data collected by the researcher or any other researcher in the future. However, all the research instruments used to collect the data are available in the appendices for researchers who would like to replicate the work. Researchers are recommended to seek ethical approval that allows them to share collected data, if possible, to increase the reproducibility and verification of their research results.

8.4.4 Future Research Directions

The previous work available on the challenges that NNES students face in programming and CS, in general, is very limited. Additionally, research that includes students in CS or programming courses rarely highlights the native language or the English proficiency level of the participants. Failing to report participants' English proficiency

levels can limit the applicability of many existing and future studies to native English speakers or those who are fluent in the language. Thus, it is recommended that all future research in the area of computer education consider reporting the English proficiency level of participants whenever applicable. This can lead to a better understanding of the impact of English from different perspectives and might highlight other language-related challenges that are still unknown.

Moreover, research on the impact of adopting EMI, especially in CS, is still limited. One recommended direction for future research is to investigate the impact of adopting EMI on the course load. The results of such research might help institutions reevaluate the number of credits that students in such courses are able to handle each semester, particularly NNES students. Another important issue that might arise from the adoption of EMI is the cognitive overload that NNES students might experience. The impact of this factor, particularly in CS1 courses, and ways to address it is a topic that deserves more attention in future research.

In the results, the majority of surveyed students believed that using English in CS1 courses represents a low to moderate barrier to their learning. The factors investigated in this study might be part of this barrier; however, a more in-depth investigation might be required to understand the students' perspective on this particular issue. This might require a qualitative research approach to explore the cause of such beliefs and their impact on the students' attitudes toward the course.

The students also expressed a desire to use a combination of English and their native language in CS1 courses. There is still no evidence in the literature that applying bilingual education in CS has a significant positive impact on students' academic performance. However, enforcing EMI in certain majors, especially in non-English-speaking countries, can potentially strip away an essential right that students might be entitled to: the right to access education in their native language.

In many other countries, students may be deprived of their basic right to engage in class, regardless of their language background, race, or culture, when EMI is used. Thus, the ethical and legal implications of adopting EMI are topics that are worth investigating. More focus might need to be placed on the equity issues in EMI courses that have both native and non-native English-speaking students enrolled.

The important question that future research needs to answer is how to provide NNES students with an equal learning experience to their native English-speaking peers while applying EMI. Investigating the impact of students' English proficiency levels on the evaluation results of their assignments and exam papers in CS courses might reveal valuable information.

Reading and understanding error messages presented one of the main challenges in the survey results, which is not limited to NNES students, as has been mentioned in other research. This highlights the need for a more collaborative effort from researchers and developers to work on enhancing the readability and usability of error messages.

Another challenge identified by NNES students in this research is the use of the English interface of coding software. Similar issues were mentioned in research conducted on block-based programming languages but not on advanced programming language coding software. More research is required to understand the actual impact of this challenge on students' ability to learn programming in advanced programming courses.

Moreover, understanding the keywords used in English-based programming languages represented a challenge for NNES students. Future research might consider investigating how this challenge can impact students' learning progress in programming courses.

Other challenges that any novice programmers might experience have also been noticed in the results, such as code reading and comprehension, problem-solving, and computational thinking. Therefore, research on this topic needs to be continued.

Some other future directions for research have already been mentioned in the discussion of the limitations of this thesis. One potential area of investigation is comparing the impact of using English and English-based programming languages on students whose native languages use the Roman alphabet versus those whose native languages use different alphabets, characters, or symbols. The similarity between English and certain European languages might impact how students who speak those languages perceive the use of English in programming courses. Another topic that might reveal interesting results is the investigation of the impact of using English-based programming languages to teach students who have no basic knowledge of English.

In terms of computer terminology, it could be interesting to explore the impact of native English-speaking students' knowledge of such terms on their academic performance in the CS1 course. This research also showed the potential for improving students' knowledge of computer terminology through a peer-reviewed translation activity; however, the impact of this activity on their academic performance in the course has not been explored. Future studies might consider investigating the benefits of applying such an activity in CS1 courses to improve students' academic performance.

This research investigated the impact of FLCA levels on the academic performance of NNES students in CS1 courses. Although the influence of this factor was found to be limited, the expanded analysis revealed interesting results in terms of gender differences. Female students showed higher levels of FLCA, but the academic performance

of male students showed a stronger correlation with the same factor. The impact of gender on FLCA and how it relates to performance in the CS1 course is a topic worth investigating. Additionally, future studies should consider other related factors, such as cultural and economic influences, and how these may affect students' acceptance of EMI and their academic performance in such courses.

The number of female students who participated in this study is significantly lower than that of male students. Although these numbers may not accurately represent the overall class demographics, as they only account for those who voluntarily participated, it is evident that CS remains imbalanced in terms of gender in many countries. Further research is necessary to explore this issue and to promote a more gender-balanced CS community.

Finally, one of the challenges for NNES students in learning programming is the availability of high-quality learning materials in languages other than English. This is a major obstacle not only for students but also for instructors and institutions searching for ways to support their NNES students. However, overcoming this problem is not an easy task and requires significant collaborative work from computer science and language specialists from different language backgrounds. Future conferences and working groups might be able to take the first step toward solving this issue, which is only achievable if enough attention and effort are dedicated to it.

8.5 Concluding Remarks

NNES students face various English-related challenges in CS1 courses that can influence their academic performance, which consequently might determine their continuation in the CS field. This important factor, along with many others, may have contributed to the global shortage of skilled programmers. Institutions, instructors, and the education system in general should work towards providing more equal opportunities for all students, regardless of their language background, to participate in the future economy.

This research represents one of the first investigations to consider different perspectives and factors aimed at understanding NNES students' struggles in their first formal experience of learning programming. It is important to note that although this research provides valuable insights, it is not comprehensive, and further investigation is still required. The major significance of this research lies in the fact that it has highlighted the importance of this topic, identified key directions for future studies, and established a solid foundation for further research in this field.

BIBLIOGRAPHY

- [1] M. Hertz, "What do "CS1" and "CS2" mean? investigating differences in the early courses," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 199–203. [Online]. Available: <https://doi.org/10.1145/1734263.1734335>
- [2] G. L. N. M. J. D. D. H. K. A. H. T. L. H. M. L. Benjamin Xie, Dastyni Loksa and A. J. Ko, "A theory of instruction for introductory programming skills," *Computer Science Education*, vol. 29, no. 2-3, pp. 205–253, 2019. [Online]. Available: <https://doi.org/10.1080/08993408.2019.1565235>
- [3] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer science education*, vol. 13, no. 2, pp. 137–172, 2003. [Online]. Available: <https://doi.org/10.1076/csed.13.2.137.14200>
- [4] Y. Qian and J. Lehman, "Students' misconceptions and other difficulties in introductory programming: A literature review," *ACM Transactions on Computing Education*, vol. 18, no. 1, Oct 2017. [Online]. Available: <https://doi.org/10.1145/3077618>
- [5] R. R. Leeper and J. L. Silver, "Predicting success in a first programming course," *SIGCSE Bulletin*, vol. 14, no. 1, pp. 147–150, February 1982. [Online]. Available: <https://doi.org/10.1145/953051.801357>
- [6] D. F. Butcher and W. A. Muth, "Predicting performance in an introductory computer science course," *Commun. ACM*, vol. 28, no. 3, pp. 263–268, March 1985. [Online]. Available: <https://doi.org/10.1145/3166.3167>
- [7] R. J. Koubek, W. K. LeBold, and G. Salvendy, "Predicting performance in computer programming courses," *Behaviour Information Technology*, vol. 4, no. 2, pp. 113–129, 1985. [Online]. Available: <https://doi.org/10.1080/01449298508901793>
- [8] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, "Analyzing and predicting students' performance by means of machine learning: A review," *Applied sciences*, vol. 10, no. 3, 2020. [Online]. Available: <https://doi.org/10.3390/app10031042>

- [9] A. Namoun and A. Alshankiti, "Predicting student performance using data mining and learning analytics techniques: A systematic literature review," *Applied Sciences*, vol. 11, no. 1, 2020. [Online]. Available: <https://doi.org/10.3390/app11010237>
- [10] P. J. Guo, "Non-native English speakers learning computer programming: Barriers, desires, and design opportunities," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, Conference Proceedings, pp. 1–14. [Online]. Available: <https://doi.org/10.1145/3173574.3173970>
- [11] Y. Pal and S. Iyer, "Classroom versus screencast for native language learners: Effect of medium of instruction on knowledge of programming," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '15. New York, NY, USA: Association for Computing Machinery, 2015, Conference Proceedings, pp. 290–295. [Online]. Available: <https://doi.org/10.1145/2729094.2742618>
- [12] H. Draga. (2023) Examples of string variables [translated from arabic]. (Last accessed September 21, 2023). [Online]. Available: <https://www.aliflang.org/docs/string>
- [13] S. Dasgupta and B. M. Hill, "Learning to code in localized programming languages," in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, ser. L@S '17. New York, NY, USA: Association for Computing Machinery, 2017, Conference Proceedings, pp. 33–39. [Online]. Available: <https://doi.org/10.1145/3051457.3051464>
- [14] T. van der Storm and F. Hermans, "Gradual grammars: Syntax in levels and locales," in *Proceedings of the 15th ACM SIGPLAN International Conference on Software Language Engineering*, ser. SLE 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 134–147. [Online]. Available: <https://doi.org/10.1145/3567512.3567524>
- [15] J. E. Sammet, *Programming Languages: History and Fundamentals*. USA: Prentice-Hall, Inc., 1969.
- [16] D. Ipacs. (2023, March) High-level programming languages: What's the secret to their coding power? Bluebird International. (Last accessed September 22, 2023). [Online]. Available: <https://bluebirdinternational.com/high-level-programming-languages/>
- [17] N. Ahmed, A. Kamal, A. Nuruddin, and S. T. Jishan, "My code in my native tone: Cha script," in *Proceedings of the Eighth International Conference on Information and Communication Technologies and Development*, ser. ICTD '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: <https://doi.org/10.1145/2909609.2909611>
- [18] J. A. S. da Costa, R. Gheyi, F. Castor, P. R. F. de Oliveira, M. Ribeiro, and B. Fonseca, "Seeing confusion through a new lens: on the impact of atoms

- of confusion on novices' code comprehension," *Empirical Software Engineering*, vol. 28, no. 4, p. 81, 2023. [Online]. Available: <https://doi.org/10.1007/s10664-023-10311-0>
- [19] (2018, August) The language of codes : Why English is the lingua franca of programming. Y Studios LLC. (Last accessed September 22, 2023). [Online]. Available: <https://ystudios.com/insights-passion/codelanguage#>
- [20] E. Macaro, *English medium instruction*. Oxford University Press, 2018.
- [21] R. Mason and C. Seton, "Leveling the playing field for international students in IT courses," in *Proceedings of the 23rd Australasian Computing Education Conference*, ser. ACE '21. New York, NY, USA: Association for Computing Machinery, 2021, Conference Proceedings, pp. 138–146. [Online]. Available: <https://doi.org/10.1145/3441636.3442316>
- [22] D. Horton and M. Craig, "Drop, fail, pass, continue: Persistence in CS1 and beyond in traditional and inverted delivery," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '15. New York, NY, USA: Association for Computing Machinery, 2015, Conference Proceedings, pp. 235–240. [Online]. Available: <https://doi.org/10.1145/2676723.2677273>
- [23] J. Ivins and M. P. Ong, "Psychometric assessment of computing undergraduates," in *proceedings of the 17th Workshop of the Psychology of Programming Interest Group*, Sussex University, 2005, pp. 305–319. [Online]. Available: <https://www.ppig.org/files/2005-PPIG-17th-ivins.pdf>
- [24] P. Kinnunen and L. Malmi, "Why students drop out CS1 course?" in *Proceedings of the Second International Workshop on Computing Education Research*, ser. ICER '06. New York, NY, USA: Association for Computing Machinery, 2006, Conference Proceedings, pp. 97–108. [Online]. Available: <https://doi.org/10.1145/1151588.1151604>
- [25] C. Watson and F. W. Li, "Failure rates in introductory programming revisited," in *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ser. ITiCSE '14. New York, NY, USA: Association for Computing Machinery, 2014, Conference Proceedings, pp. 39–44. [Online]. Available: <https://doi.org/10.1145/2591708.2591749>
- [26] R. A. Alturki *et al.*, "Measuring and improving student performance in an introductory programming course," *Informatics in Education-An International Journal*, vol. 15, no. 2, pp. 183–204, 2016. [Online]. Available: <https://www.ceeol.com/search/article-detail?id=442940>
- [27] E. Lindoo, "Back to the basics in an effort to improve student retention in intro to programming classes," *Journal of Computing Sciences in Colleges*, vol. 34, no. 2, pp. 72–79, December 2018. [Online]. Available: <https://dl.acm.org/doi/abs/10.5555/3282588.3282599>

- [28] M. Naseem, K. Chaudhary, B. Sharma, and A. G. Lal, "Using ensemble decision tree model to predict student dropout in computing science," in *IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2019, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/CSDE48274.2019.9162389>
- [29] N. Pillay and V. R. Jugoo, "An investigation into student characteristics affecting novice programming performance," *SIGCSE Bulletin*, vol. 37, no. 4, pp. 107–110, December 2005. [Online]. Available: <https://doi.org/10.1145/1113847.1113888>
- [30] M. M. Mhashi and A. Alakeel, "Difficulties facing students in learning computer programming skills at Tabuk University," *Recent Advances in Modern Educational Technologies*, pp. 15–24, 2013. [Online]. Available: <https://www.tib.eu/en/search/id/BLCP%3ACN084897952/Difficulties-Facing-Students-in-Learning-Computer/>
- [31] M. Ben Idris and H. Ammar, "The correlation between Arabic student's English proficiency and their computer programming ability at the university level," *International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT)*, vol. 9, 2018. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.3466182>
- [32] K. J. Keen and L. Etzkorn, "Predicting students' grades in computer science courses based on complexity measures of teacher's lecture notes," *Journal of Computing Sciences in Colleges*, vol. 24, no. 5, pp. 44–48, May 2009.
- [33] A. Masrai, D. A. Salam El-Dakhs, and H. Al Khawar, "General and specialist vocabulary knowledge as predictors of academic success in EMI university programs," *Language Learning in Higher Education*, vol. 12, no. 1, pp. 273–289, 2022. [Online]. Available: <https://doi.org/10.1515/cercles-2022-2038>
- [34] C. A. Lager, "Types of mathematics-language reading interactions that unnecessarily hinder algebra learning and assessment," *Reading Psychology*, vol. 27, no. 2-3, pp. 165–204, 2006. [Online]. Available: <https://doi.org/10.1080/02702710600642475>
- [35] E. W. Dijkstra, "How do we tell truths that might hurt?" *SIGPLAN Notices*, vol. 17, no. 5, pp. 13–15, May 1982. [Online]. Available: <https://doi-org.ucd.idm.oclc.org/10.1145/947923.947924>
- [36] A. G. Applin, "Second language acquisition and CS1," in *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '01. New York, NY, USA: Association for Computing Machinery, 2001, pp. 174–178. [Online]. Available: <https://doi.org/10.1145/364447.364579>
- [37] Y. Qian, P. Yan, and M. Zhou, "Using data to understand difficulties of learning to program: A study with Chinese middle school students," in *Proceedings of the ACM Conference on Global Computing Education*, ser. CompEd '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 185–191. [Online]. Available: <https://doi.org/10.1145/3300115.3309521>

- [38] A. A. Khuwaileh, "IT terminology and translation: Cultural, lexicographic and linguistic problems," *LSP Journal-Language for special purposes, professional communication, knowledge management and cognition*, vol. 1, no. 2, pp. 15–22, 2010. [Online]. Available: <https://rauli.cbs.dk/index.php/lspcog/article/view/3089>
- [39] D. Bouvier, E. Lovellette, J. Matta, B. Alshaigy, B. A. Becker, M. Craig, J. Jackova, R. McCartney, K. Sanders, and M. Zarb, "Novice programmers and the problem description effect," in *Proceedings of the 2016 ITiCSE Working Group Reports*, ser. ITiCSE '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 103–118. [Online]. Available: <https://doi.org/10.1145/3024906.3024912>
- [40] L. Carter, L. Jernejcic, and N. Lim, "Success in CS: Is culture a factor?" in *37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, 2007, pp. T3A–16–T3A–21. [Online]. Available: <https://doi.org/10.1109/FIE.2007.4417819>
- [41] R. Mason and C. Seton, "Assessing international students: The role of cognitive load," in *Proceedings of the Twenty-Second Australasian Computing Education Conference*, ser. ACE'20. New York, NY, USA: Association for Computing Machinery, 2020, Conference Proceedings, pp. 160–166. [Online]. Available: <https://doi.org/10.1145/3373165.3373183>
- [42] N. Suzuki, "Foreign language speaking anxiety in the English-medium instruction (EMI) program in a Japanese university," *Casele Research Bulletin*, vol. 47, pp. 11–21, 2017. [Online]. Available: https://doi.org/10.18983/casele.47.0_11
- [43] W. A. Suliman and A. Tadros, "Nursing students coping with english as a foreign language medium of instruction," *Nurse Education Today*, vol. 31, no. 4, pp. 402–407, 2011. [Online]. Available: <https://doi.org/10.1016/j.nedt.2010.07.014>
- [44] S. Chun, H. Kim, C.-K. Park, K. McDonald, O. Sun Ha, D. L. Kim, and S. M. Lee, "South Korean students' responses to English-medium instruction courses," *Social Behavior and Personality: an international journal*, vol. 45, no. 6, pp. 951–965, 2017. [Online]. Available: <https://doi.org/10.2224/sbp.6049>
- [45] L. Rohliah, P. Lestari, W. Ningsih, and G. U. Hutagalung, "Non-English major students' foreign language anxiety: A study case," *Global Expert: Jurnal Bahasa dan Sastra*, vol. 11, no. 1, pp. 7–16, 2023. [Online]. Available: <https://doi.org/10.36982/jge.v11i1.3201>
- [46] Y. Teimouri, J. Goetze, and L. Plonsky, "Second language anxiety and achievement: A meta-analysis," *Studies in Second Language Acquisition*, vol. 41, no. 2, pp. 363–387, 2019. [Online]. Available: <https://doi.org/10.1017/S0272263118000311>
- [47] J. Knight, "The changing landscape of higher education internationalisation - for better or worse?" *Perspectives: Policy and Practice in Higher Education*, vol. 17, no. 3, pp. 84–90, 2013. [Online]. Available: <https://doi.org/10.1080/13603108.2012.753957>

- [48] C. A. García, S. Weale, L. Swan, and H. Symons. (2023, July) Fifth of UK universities' income comes from overseas students, figures show. (Last accessed September 22, 2023). [Online]. Available: <https://www.theguardian.com/education/2023/jul/14/overseas-students-uk-universities-income>
- [49] (2023, May) International students boost UK economy by £41.9 billion. (Last accessed September 22, 2023). [Online]. Available: <https://www.universitiesuk.ac.uk/latest/news/international-students-boost-uk-economy>
- [50] A. Smith. (2023, February) The software developer shortage and what we can do about it. (Last accessed September 19, 2023). [Online]. Available: <https://hatchworks.com/the-software-developer-shortage-and-what-we-can-do-about-it/>
- [51] M. Franzino, A. Guarino, Y. Binvel, and J.-M. Laouchez. (2023) The \$ 8.5 trillion talent shortage. (Last accessed September 19, 2023). [Online]. Available: www.kornferry.com/insights/this-week-in-leadership/talent-crunch-future-of-work
- [52] I. Ruby and B. Krsmanovic, "Does learning a programming language require learning English? a comparative analysis between English and programming languages," in *Proceedings of EdMedia + Innovate Learning 2017*, J. P. Johnston, Ed. Washington, DC: Association for the Advancement of Computing in Education (AACE), June 2017, pp. 420–427. [Online]. Available: <https://www.learntechlib.org/p/178486>
- [53] W. W. Lau and A. H. Yuen, "Modelling programming performance: Beyond the influence of learner characteristics," *Computers Education*, vol. 57, no. 1, pp. 1202–1213, 2011. [Online]. Available: <https://doi.org/10.1016/j.compedu.2011.01.002>
- [54] S. Zusana E. Pudyastuti and M. Jozua F. Palandi, "A correlation between students English proficiency and their computer programming mastery," in *the 3rd UAD TEFL International Conference*, vol. 3, 2014, Conference Paper, pp. 1219–1228.
- [55] S. Rauchas, B. Rosman, G. Konidaris, and I. Sanders, "Language performance at high school and success in first year computer science," in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '06. New York, NY, USA: Association for Computing Machinery, 2006, Conference Proceedings, pp. 398–402. [Online]. Available: <https://doi.org/10.1145/1121341.1121467>
- [56] T. Wibowo and R. Tandiono, "Correlation analysis between English language proficiency and ease of learning programming language: A case study on students of the faculty of computer science University of International Batam," in *Conference on Management, Business, Innovation, Education and Social Sciences (CoMBInES)*, vol. 2, no. 1, 2022, pp. 237–244. [Online]. Available: <https://journal.uib.ac.id/index.php/combines/article/view/6622>
- [57] M. Alhamami, "English as a medium of instruction (EMI) in computing undergraduate programs," *Education and Information Technologies*, vol. 26, no. 5, pp. 6549–6562, 2021. [Online]. Available: <https://doi.org/10.1007/s10639-021-10637-6>

- [58] H. Alrasheed, A. Alnashwan, and R. Alshowiman, "Impact of English proficiency on academic performance of software engineering students," in *the 4th International Conference on Data Storage and Data Engineering*, ser. DSDE '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 107–111. [Online]. Available: <https://doi.org/10.1145/3456146.3456163>
- [59] J. M. Nash, "Relating IS student throughput to English proficiency," in *the 36th Annual Conference of the Southern African Computer Lecturers Association: Conference Proceedings*, J.-P. V. Belle and I. Brown, Eds. University of Cape Town, 2006, pp. 21–31. [Online]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6d9d07cf733ec92b8f599caf2a52c58b7281a7a4#page=27>
- [60] Y. Qian and J. D. Lehman, "Correlates of success in introductory programming: A study with middle school students," *Journal of Education and Learning*, vol. 5, no. 2, pp. 73–83, 2016. [Online]. Available: <http://dx.doi.org/10.5539/jel.v5n2p73>
- [61] I. Gülmez and N. Özdener, "Academic achievement in computer programming instruction and effects of the use of visualization tools; at the elementary school level," *British Journal of Education, Society and Behavioural Science*, vol. 11, no. 1, pp. 1–18, 2015. [Online]. Available: <https://doi.org/10.9734/BJESBS/2015/18316>
- [62] O. T. Virkki, "Computer science student selection - a scoping review and a national entrance examination reform," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 654–659. [Online]. Available: <https://doi.org/10.1145/3408877.3432371>
- [63] R. H. Nowaczyk, "Cognitive skills needed in computer programming," 1983. [Online]. Available: <https://www.learntechlib.org/p/136288>
- [64] T.-S. Fan, Y.-C. Li, and M. L. Niess, "Predicting academic achievement of college computer science majors," *Journal of Research on Computing in Education*, vol. 31, no. 2, pp. 155–172, 1998. [Online]. Available: <https://doi.org/10.1080/08886504.1998.10782248>
- [65] M. Anwar, N. Ahmed, and W. Khan, "Analysis of students' grades in Mathematics, English, and Programming courses: A KDD approach," *International Journal of Future Computer and Communication*, vol. 1, no. 2, pp. 111–115, 2012.
- [66] E. Mauai and E. Temese, "Exploratory study on factors that impact/influence success and failure of students in the foundation computer studies course at the National University of Samoa," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 5, pp. 767–773, 2012.
- [67] L. Mutanu and P. Machoka, "Enhancing computer students' academic performance through predictive modelling - a proactive approach," in *the 14th International Conference on Computer Science & Education (ICCSE)*, 2019, pp. 97–102. [Online]. Available: <https://doi.org/10.1109/ICCSE.2019.8845452>

- [68] M. Keane and G. Gray, "An investigation into the pre-enrolment characteristics of students to identify factors predictive of academic performance within first year computing and engineering programmes of study in a higher educational institution," *All Ireland Journal of Higher Education*, vol. 11, no. 1, 2019. [Online]. Available: <https://ojs.aishe.org/index.php/aishe-j/article/view/382>
- [69] G. A. Chmura, "What abilities are necessary for success in computer science," *SIGCSE Bulletin*, vol. 30, no. 4, pp. 55–58, December 1998. [Online]. Available: <https://doi-org.ucd.idm.oclc.org/10.1145/306286.306316>
- [70] P. Byrne and G. Lyons, "The effect of student attributes on success in programming," in *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '01. New York, NY, USA: Association for Computing Machinery, 2001, Conference Proceedings, pp. 49–52. [Online]. Available: <https://doi.org/10.1145/377435.377467>
- [71] C. Blewett and D. Achmad, "Towards identifying predictors of student performance in learning computer programming at a South African University," in *Proceedings of the 35th Conference of SACLA*, Kasane, Botswana, July 2005, Conference Proceedings, pp. 57–63.
- [72] G. Barlow-Jones and D. van der Westhuizen, "The correlation between a students English proficiency and their grasp of computer programming," in *Proceedings of Global Learn-Global Conference on Learning and Technology*. Limerick, Ireland: Association for the Advancement of Computing in Education (AACE), April 2016, Conference Proceedings, pp. 569–573. [Online]. Available: <https://www.learntechlib.org/p/172789>
- [73] —, "Pre-entry attributes thought to influence the performance of students in computer programming," in *ICT Education*, J. Liebenberg and S. Gruner, Eds. Springer International Publishing, 2017, Conference Proceedings, pp. 217–226.
- [74] Y. Ayalew, E. Tshukudu, and M. Lefoane, "Factors affecting programming performance of first year students at a university in Botswana," *African Journal of Research in Mathematics, Science and Technology Education*, vol. 22, no. 3, pp. 363–373, 2018. [Online]. Available: <https://hdl.handle.net/10520/EJC-12a2c586fa>
- [75] L. J. Mselle and H. Twaakyondo, "The impact of memory transfer language (MTL) on reducing misconceptions in teaching programming to novices," *International Journal of Machine Learning and Applications*, vol. 1, no. 1, p. 6, 2012. [Online]. Available: <https://doi.org/10.4102/ijmla.v1i1.3>
- [76] H. Chan, "Pre-teaching technical vocabulary in an introductory information systems course: An experiment involving non-native speakers of English," in *Proceedings of the Information Resources Management Association International Conference on Challenges of Information Technology Management in the 21st Century*. USA: IGI Global, 2000, Conference Proceedings, pp. 74–78.
- [77] H. Gedawy, S. Razak, and H. Alshikhabobakr, "The effectiveness of creating localized content for middle school computing curriculum," in *Proceedings*

- of the ACM Conference on Innovation and Technology in Computer Science Education, ser. ITiCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, Conference Proceedings, pp. 478–484. [Online]. Available: <https://doi.org/10.1145/3304221.3319778>
- [78] A. Espinal, C. Vieira, and V. Guerrero-Bequis, “Student ability and difficulties with transfer from a block-based programming language into other programming languages: a case study in Colombia,” *Computer Science Education*, pp. 1–33, 2022. [Online]. Available: <https://doi.org/10.1080/08993408.2022.2079867>
- [79] S. Gökoğlu and S. Kilic, “Programming learning and teaching of pre-service computer science teachers: Challenges, concerns, and solutions,” *E-Learning and Digital Media*, pp. 498–518, 2022. [Online]. Available: <https://doi.org/10.1177/20427530221117331>
- [80] M. T. Cabré, *Terminology: Theory, methods, and applications*. John Benjamins Publishing, 1999, vol. 1.
- [81] G. Sauberer, “There is no knowledge without terminology. how terminological methods and tools can help to manage monolingual and multilingual knowledge and communication,” *Systems, Cybernetics and Informatics*, vol. 8, no. 2, pp. 56–60, 2011. [Online]. Available: <https://www.iiisci.org/journal/sci/Contents.asp?var=&Previous=ISS0908>
- [82] G. M. Bakhretdinovna, “Use of English terminology in technical universities,” *American Journal of Interdisciplinary Research and Development*, vol. 16, pp. 1–4, 2023. [Online]. Available: <https://www.ajird.journalspark.org/index.php/ajird/article/view/647>
- [83] W. Hasselbring, “Technical opinion: On defining computer science terminology,” *Communications of the ACM*, vol. 42, no. 2, pp. 88–91, 1999. [Online]. Available: <https://dl.acm.org/doi/fullHtml/10.1145/293411.293488>
- [84] A. Snyder, “The essence of objects: concepts and terms,” *IEEE Software*, vol. 10, no. 1, pp. 31–42, 1993.
- [85] S. A. Ward, “The hierarchical terminology technique: a method to address terminology inconsistency,” *Qual Quant*, vol. 46, pp. 71–87, 2012. [Online]. Available: <https://doi.org/10.1007/s11135-010-9328-6>
- [86] I. Diethelm and J. Goschler, “Questions on spoken language and terminology for teaching computer science,” in *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 21–26. [Online]. Available: <https://doi.org/10.1145/2729094.2742600>
- [87] A. C. Danto, “Metaphor and cognition,” in *Knowledge and Language: Volume III Metaphor and Knowledge*. Springer Dordrecht, 1993, pp. 21–35.

- [88] T. Colburn and G. Shute, "Metaphor in computer science," *Journal of Applied Logic*, vol. 6, no. 4, pp. 526–533, 2008, the Philosophy of Computer Science. [Online]. Available: <https://doi.org/10.1016/j.jal.2008.09.005>
- [89] R. Moini and Z. Islamizadeh, "Do we need discipline-specific academic word lists? linguistics academic word list (LAWL)," *Journal of Teaching Language Skills*, vol. 35, no. 3, pp. 65–90, 2016. [Online]. Available: <https://doi.org/10.22099/jtls.2016.3901>
- [90] N. Kern, "Technology-integrated English for specific purposes lessons: real-life language, tasks, and tools for professionals," in *Innovations in learning technologies for English language teaching*, G. Motteram, Ed. British Council, 2013.
- [91] B. Munasinghe, T. Bell, and A. Robins, "Teachers's understanding of technical terms in a computational thinking curriculum," in *Proceedings of the 23rd Australasian Computing Education Conference*, ser. ACE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 106â114. [Online]. Available: <https://doi.org/10.1145/3441636.3442311>
- [92] Y. Pal, "A framework for scaffolding to teach programming to vernacular medium learners," PhD thesis, Indian Institute of Technology Bombay, 2016.
- [93] D. E. Minshall, "A computer science word list," Masters dissertation, 2013. [Online]. Available: <https://www.baleap.org/wp-content/uploads/2016/03/Daniel-Minshall.pdf>
- [94] B. Scott and R. Mason, "Cyber as a second language? a challenge to cybersecurity education," in *Journal of The Colloquium for Information Systems Security Education*, vol. 9, no. 1, 2022, pp. 6–6. [Online]. Available: <https://doi.org/10.53735/cisse.v9i1.137>
- [95] L. Benchekroune, "English for IT purposes: Moroccan ethical hackers as a case study," *International Journal of English Language Studies*, vol. 2, pp. 39–43, 2020. [Online]. Available: <https://doi.org/10.32996/ijels.2020.2.5.5>
- [96] L. Q. Dong, P. T. N. Le, and N. T. Tuyet, "EMI for engineering-technology students in vietnam: Challenges and suggestions," *VNU Journal of Foreign Studies*, vol. 38, no. 5, pp. 53–74, 2022. [Online]. Available: <https://jfs.ulis.vnu.edu.vn/index.php/fs/article/view/5072>
- [97] L. Gawazah and H. Z. Woldemariam, "Interrogating the contemporary English language needs for the ICT industry in the Namibian context," 2023. [Online]. Available: <http://ir.nust.na:8080/jspui/handle/10628/985>
- [98] Y. Ehara, "Analyzing semantically distinctive english word usages in computer science for english-as-a-second-language learners," in *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2*, ser. ITiCSE '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 636. [Online]. Available: <https://doi-org.ucd.idm.oclc.org/10.1145/3502717.3532122>

- [99] M. Kourichi and R. Benyelles, "English for computer science: Students' needs to enhance technical writing at Tlemcen University," pp. 113–130, 2019. [Online]. Available: <https://aleph.edinum.org/1825>
- [100] B. A. Becker, "The roles of computing terminology in non-computing disciplines," in *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 2*, ser. ITiCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 659. [Online]. Available: <https://doi-org.ucd.idm.oclc.org/10.1145/3456565.3460073>
- [101] S. Esche and K. Weihe, "Case study on the terms novice programmers use to describe code snippets in Java," *IEEE Transactions on Education*, vol. 66, no. 6, pp. 642–653, 2023.
- [102] J. Rountree, N. Rountree, A. Robins, and R. Hannah, "Observations of student competency in a CS1 course," in *Proceedings of the 7th Australasian Conference on Computing Education*, ser. ACE '05, vol. 42. AUS: Australian Computer Society, Inc., 2005, Conference Proceedings, pp. 145–149. [Online]. Available: dl.acm.org/doi/abs/10.5555/1082424.1082442
- [103] P. Rein, M. Taeumel, and R. Hirschfeld, *Towards a Theory of Factors that Influence Text Comprehension of Code Documents*. Cham: Springer International Publishing, 2021, pp. 307–325. [Online]. Available: https://doi.org/10.1007/978-3-030-62037-0_14
- [104] F. Hermans, A. Swidan, and E. Aivaloglou, "Code phonology: An exploration into the vocalization of code," in *Proceedings of the 26th Conference on Program Comprehension*, ser. ICPC '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 308–311. [Online]. Available: <https://doi.org/10.1145/3196321.3196355>
- [105] M. Mladenović, M. Rosić, and S. Mladenović, "Comparing elementary students' programming success based on programming environment," *International Journal of Modern Education and Computer Science*, vol. 8, no. 8, pp. 1–10, 2016. [Online]. Available: <https://doi.org/10.5815/ijmecs.2016.08.01>
- [106] A. Stefik and S. Siebert, "An empirical investigation into programming language syntax," *ACM Transactions on Computing Education (TOCE)*, vol. 13, no. 4, pp. 1–40, November 2013. [Online]. Available: <https://doi.org/10.1145/2534973>
- [107] V. Lappi, V. Tirronen, and J. Itkonen, "A replication study on the intuitiveness of programming language syntax," *Software Quality Journal*, pp. 1–30, 2023. [Online]. Available: <https://doi.org/10.1007/s11219-023-09631-7>
- [108] M. Clancy, "Misconceptions and attitudes that interfere with learning to program," in *Computer science education research*, M. P. Sally Fincher, Ed. Taylor / Francis, 2004, pp. 85–100.
- [109] D. Lawrie, C. Morrell, H. Feild, and D. Binkley, "What's in a name? a study of identifiers," in *14th IEEE International Conference on Program*

- Comprehension (ICPC'06)*, 2006, Conference Proceedings, pp. 3–12. [Online]. Available: <https://doi.org/10.1109/ICPC.2006.51>
- [110] H. Elazhary, “Facile programming,” *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 256–261, 2012. [Online]. Available: <https://iajit.org/PDF/vol.9,no.3/2717-9.pdf>
- [111] M. Gilsing, J. Pelay, and F. Hermans, “Design, implementation and evaluation of the hedy programming language,” *Journal of Computer Languages*, vol. 73, p. 101158, 2022. [Online]. Available: <https://doi.org/10.1016/j.cola.2022.101158>
- [112] E. K. Olatunji, J. B. Oladosu, O. A. Odejobi, and S. O. Olabiyisi, “A needs assessment for indigenous African language-based programming languages,” *Annals of Science and Technology*, vol. 4, no. 2, pp. 1–5, 2019. [Online]. Available: <https://sciendocom/downloadpdf/journals/ast/4/2/article-p1.pdf>
- [113] G. Obaido, A. Ade-Ibijola, and H. Vadapalli, “Synthesis of SQL queries from South African local language narrations,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 1189–1195, 2020. [Online]. Available: <https://doi.org/10.25046/aj0505144>
- [114] G. Silva, G. Santos, E. D. Canedo, V. Rissoli, B. Praciano, and G. Andrade, “Impact of Calango language in an introductory computer programming course,” in *IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–9. [Online]. Available: <https://doi.org/10.1109/FIE44824.2020.9274150>
- [115] B. Goswami and S. Pal, “Introduction of two new programming tools in Bengali and measurement of their reception among high-school students in Purba Bardhaman, India with the prototypic inclusion of a vector-biology module,” *Education and Information Technologies*, pp. 1–23, 2022. [Online]. Available: <https://doi.org/10.1007/s10639-021-10663-4>
- [116] M. Rahmat, S. Shahrani, R. Latih, N. F. M. Yatim, N. F. A. Zainal, and R. Ab Rahman, “Major problems in basic programming that influence student performance,” *Procedia-Social and Behavioral Sciences*, vol. 59, pp. 287–296, 2012. [Online]. Available: <https://doi.org/10.1016/j.sbspro.2012.09.277>
- [117] A. E. Elgobshaw, “Translation and Arabization of computer terminology: A study of learners’ preferences and attitudes,” *Eurasian Journal of Applied Linguistics*, vol. 8, no. 1, pp. 313–322, 2022. [Online]. Available: <https://ejal.info/menuscrypt/index.php/ejal/article/view/196>
- [118] C. N. Guzman, A. Xu, and A. Gerald Soosai Raj, “Experiences of Non-Native English Speakers Learning Computer Science in a US University,” in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 633–639. [Online]. Available: <https://doi.org/10.1145/3408877.3432437>
- [119] Stackoverflow. (2019) Developer survey results 2019. (Last accessed August 30, 2023). [Online]. Available: <https://insights.stackoverflow.com/survey/2019>

- [120] B. Xu, Z. Xing, X. Xia, D. Lo, Q. Wang, and S. Li, "Domain-specific cross-language relevant question retrieval," in *Proceedings of the 13th International Conference on Mining Software Repositories*, ser. MSR '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 413–424. [Online]. Available: <https://doi.org/10.1145/2901739.2901746>
- [121] P. J. Rich, S. Bartholomew, D. Daniel, K. Dinsmoor, M. Nielsen, C. Reynolds, M. Swanson, E. Winward, and J. Yauney, "Trends in tools used to teach computational thinking through elementary coding," *Journal of Research on Technology in Education*, pp. 1–22, 2022. [Online]. Available: <https://doi.org/10.1080/15391523.2022.2121345>
- [122] P. G. Feijóo-García, K. McNamara, and J. Stuart, "Work in progress: Native language consistency effect on learners' introduction to block-based instructional technologies," in *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, ser. SIGITE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 147. [Online]. Available: <https://doi.org/10.1145/3349266.3351368>
- [123] B. A. Becker, P. Denny, R. Pettit, D. Bouchard, D. J. Bouvier, B. Harrington, A. Kamil, A. Karkare, C. McDonald, P.-M. Osera, J. L. Pearce, and J. Prather, "Compiler error messages considered unhelpful: The landscape of text-based programming error message research," in *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*, ser. ITiCSE-WGR '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 177–210. [Online]. Available: <https://doi.org/10.1145/3344429.3372508>
- [124] B. A. Becker, "Parlez-vous Java? Bonjour La Monde!= Hello World: Barriers to programming language acquisition for non-native English speakers." in *the 30th Annual Conference of the Psychology of Programming Interest Group (PPIG '19)*, 2019, pp. 40–51. [Online]. Available: <https://www.ppig.org/files/2019-PPIG-30th-becker.pdf>
- [125] K. Reestman and B. Dorn, "Native language's effect on Java compiler errors," in *Proceedings of the ACM Conference on International Computing Education Research*, ser. ICER '19. New York, NY, USA: Association for Computing Machinery, 2019, Conference Proceedings, pp. 249–257. [Online]. Available: <https://doi.org/10.1145/3291279.3339423>
- [126] Z. Liang, M. Nishi, and I. Kishida, "Teaching android app development to first year undergraduates: Textual programming or visual programming?" in *IEEE International Conference on Engineering, Technology & Education (TALE)*. IEEE, 2021, pp. 01–08. [Online]. Available: <https://doi.org/10.1109/TALE52509.2021.9678602>
- [127] P. Denny, J. Prather, B. A. Becker, C. Mooney, J. Homer, Z. C. Albrecht, and G. B. Powell, "On designing programming error messages for novices: Readability and its constituent factors," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ser. CHI '21. New York, NY, USA:

- Association for Computing Machinery, 2021, pp. 1–15. [Online]. Available: <https://doi.org/10.1145/3411764.3445696>
- [128] I. Nnass, “Identifying and solving issues with acquiring skills in computer programming for non-English speakers,” PhD thesis, CQUniversity, December 2019. [Online]. Available: https://acquire.cqu.edu.au/articles/thesis/Identifying_and_solving_issues_with_acquiring_skills_in_computer_programming_for_non-English_speakers/13406825
 - [129] I. Nnass, M. A. Cowling, and R. Hadgraft, “Identifying the difficulties of learning programming for non-English speakers at CQUniversity and Sebha University,” *Journal of Pure & Applied Sciences*, vol. 21, no. 4, pp. 290–295, 2022. [Online]. Available: <https://doi.org/10.51984/jopas.v21i4.2258>
 - [130] B. A. Becker, “An effective approach to enhancing compiler error messages,” in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 126–131. [Online]. Available: <https://doi.org/10.1145/2839509.2844584>
 - [131] G. Carvalho, J. C. Rocha, A. Gomes, A. J. Mendes *et al.*, “Enhanced compiler messages of error in Python with focuses in readability in CS1,” in *the Latin American Conference on Learning Technologies (LACLO)*, 2021, pp. 389–396. [Online]. Available: <https://doi.org/10.1109/LACLO54177.2021.00048>
 - [132] J. Prather, P. Denny, B. A. Becker, R. Nix, B. N. Reeves, A. S. Randrianasolo, and G. Powell, “First steps towards predicting the readability of programming error messages,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE 2023, vol. 1. New York, NY, USA: Association for Computing Machinery, 2023, pp. 549–555. [Online]. Available: <https://doi.org/10.1145/3545945.3569791>
 - [133] R. Garcia, C.-J. Liao, and A. Pearce, “Read the debug manual: A debugging manual for CS1 students,” in *IEEE Frontiers in Education Conference (FIE)*, 2022, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/FIE56618.2022.9962675>
 - [134] J. Leinonen, A. Hellas, S. Sarsa, B. Reeves, P. Denny, J. Prather, and B. A. Becker, “Using large language models to enhance programming error messages,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE 2023, vol. 1. New York, NY, USA: Association for Computing Machinery, 2023, pp. 563–569. [Online]. Available: <https://doi.org/10.1145/3545945.3569770>
 - [135] N. Eckberg, A. M. Pidgeon, and H. Magyar, “Examining the psychosocial and academic factors predicting depression and anxiety symptomology across first year and later year university students,” *European Scientific Journal*, vol. 13, no. 17, pp. 1–16, 2017.
 - [136] K. Nolan and S. Bergin, “The role of anxiety when learning to program: A systematic review of the literature,” in *Proceedings of the 16th Koli Calling*

- International Conference on Computing Education Research*, ser. Koli Calling '16. New York, NY, USA: Association for Computing Machinery, 2016, Conference Proceedings, pp. 61–70. [Online]. Available: <https://doi.org/10.1145/2999541.2999557>
- [137] S. E. Chang, "Computer anxiety and perception of task complexity in learning programming-related skills," *Computers in Human Behavior*, vol. 21, no. 5, pp. 713–728, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S074756320400038X>
- [138] M. J. Scott and G. Ghinea, "Measuring enrichment: The assembly and validation of an instrument to assess student self-beliefs in CS1," in *Proceedings of the Tenth Annual Conference on International Computing Education Research*, ser. ICER '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 123â130. [Online]. Available: <https://doi.org/10.1145/2632320.2632350>
- [139] J. Owolabi, P. Olanipekun, and J. Iwerima, "Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming," *GSTF Journal on Computing (JoC)*, vol. 3, no. 4, p. 109, 2014. [Online]. Available: <https://doi.org/10.7603/s40601-013-0047-4>
- [140] E. Doyle, I. Stamouli, and M. Huggard, "Computer anxiety, self-efficacy, computer experience: An investigation throughout a computer science degree," in *Proceedings Frontiers in Education 35th Annual Conference*. IEEE, 2005, Conference Proceedings. [Online]. Available: <https://doi.org/10.1109/FIE.2005.1612246>
- [141] P. D. MacIntyre and R. C. Gardner, "The subtle effects of language anxiety on cognitive processing in the second language," *Language learning*, vol. 44, no. 2, pp. 283–305, 1994. [Online]. Available: <https://doi.org/10.1111/j.1467-1770.1994.tb01103.x>
- [142] E. K. Horwitz, M. B. Horwitz, and J. Cope, "Foreign language classroom anxiety," *The Modern language journal*, vol. 70, no. 2, pp. 125–132, 1986. [Online]. Available: <https://doi.org/10.2307/327317>
- [143] Z. Na, "A study of high school students' english learning anxiety," *The Asian EFL Journal*, vol. 9, no. 3, pp. 22–34, 2007. [Online]. Available: https://asian-efl-journal.com/September_2007_EBook_editions.pdf#page=22
- [144] R. Cheng and A. Erben, "Language anxiety: Experiences of chinese graduate students at us higher institutions," *Journal of Studies in International Education*, vol. 16, no. 5, pp. 477–497, 2012. [Online]. Available: <https://doi.org/10.1177/1028315311421841>
- [145] V. Agarwal, Y. Chuengsatiansup, E. Kim, Y. LYu, and A. G. Soosai Raj, "An analysis of stress and sense of belonging among native and non-native English speakers learning computer science," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, vol. 1. New York, NY, USA:

- Association for Computing Machinery, 2022, pp. 376–382. [Online]. Available: <https://doi.org/10.1145/3478431.3499375>
- [146] T. M. Khan and S. W. Nabi, “English versus native language for higher education in computer science: A pilot study,” in *Proceedings of the 21st Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/10.1145/3488042.3488070>
 - [147] M.-H. Chou, “Investigating the mediating role of foreign language anxiety in studentsâ task goal orientations, perceived communication competence, english ability, and willingness to communicate in EMI classrooms,” *English Teaching & Learning*, vol. 46, no. 3, pp. 235–254, 2022.
 - [148] T. Van Gog and F. Paas, *Cognitive Load Measurement*. Boston, MA: Springer US, 2012, pp. 599–601. [Online]. Available: https://doi.org/10.1007/978-1-4419-1428-6_412
 - [149] B. B. Morrison, B. Dorn, and M. Guzdial, “Measuring cognitive load in introductory CS: Adaptation of an instrument,” in *Proceedings of the Tenth Annual Conference on International Computing Education Research*, ser. ICER '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 131–138. [Online]. Available: <https://doi.org/10.1145/2632320.2632348>
 - [150] W. Schnotz and C. Kürschner, “A reconsideration of cognitive load theory,” *Educational psychology review*, vol. 19, no. 4, pp. 469–508, 2007. [Online]. Available: <https://doi.org/10.1007/s10648-007-9053-4>
 - [151] J. J. Van Merriënboer and J. Sweller, “Cognitive load theory and complex learning: Recent developments and future directions,” *Educational psychology review*, vol. 17, no. 2, pp. 147–177, 2005. [Online]. Available: <https://doi.org/10.1007/s10648-005-3951-0>
 - [152] J. Leppink, F. Paas, C. P. Van der Vleuten, T. Van Gog, and J. J. Van Merriënboer, “Development of an instrument for measuring different types of cognitive load,” *Behavior research methods*, vol. 45, no. 4, pp. 1058–1072, 2013. [Online]. Available: <https://doi.org/10.3758/s13428-013-0334-1>
 - [153] A. Zavgorodniaia, R. Duran, A. Hellas, O. Seppala, and J. Sorva, “Measuring the cognitive load of learning to program: A replication study,” in *United Kingdom & Ireland Computing Education Research Conference.*, ser. UKICER '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 3–9. [Online]. Available: <https://doi.org/10.1145/3416465.3416468>
 - [154] A. Kaur and K. K. Chahal, “A review on the impact of cognitive factors in introductory programming,” in *Proceedings of Fourth International Conference on Communication, Computing and Electronics Systems*, V. Bindhu, J. M. R. S. Tavares, and C. Vuppapapati, Eds. Singapore: Springer Nature Singapore, 2023, pp. 1019–1032. [Online]. Available: https://doi.org/10.1007/978-981-19-7753-4_77

- [155] B. L. Kurtz, "Investigating the relationship between the development of abstract reasoning and performance in an introductory programming class," in *Proceedings of the Eleventh SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '80. New York, NY, USA: Association for Computing Machinery, 1980, Conference Proceedings, pp. 110–117. [Online]. Available: <https://doi.org/10.1145/800140.804622>
- [156] M. Ford and S. Venema, "Assessing the success of an introductory programming course," *Journal of Information Technology Education: Research*, vol. 9, no. 1, pp. 133–145, January 2010. [Online]. Available: <https://www.learntechlib.org/p/111360>
- [157] J. Sheard, A. Carbone, S. Markham, A. J. Hurst, D. Casey, and C. Avram, "Performance and progression of first year ICT students," in *Proceedings of the Tenth Conference on Australasian Computing Education*, ser. ACE '08, vol. 78. AUS: Australian Computer Society, Inc., 2008, Conference Proceedings, pp. 119–127.
- [158] J. Johnson, S. Lubo, N. Yedla, J. Aponte, and B. Sharif, "An empirical study assessing source code readability in comprehension," in *IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2019, pp. 513–523. [Online]. Available: <https://doi.org/10.1109/ICSME.2019.00085>
- [159] A. Begel and S. L. Graham, "Spoken programs," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*, 2005, pp. 99–106. [Online]. Available: <https://doi.org/10.1109/VLHCC.2005.58>
- [160] L. Guerrouj, M. Di Penta, Y.-G. Guéhéneuc, and G. Antoniol, "An experimental investigation on the effects of context on source code identifiers splitting and expansion," *Empirical Software Engineering*, vol. 19, pp. 1706–1753, 2014. [Online]. Available: <https://doi.org/10.1007/s10664-013-9260-1>
- [161] X. Li and C. Prasad, "Effectively teaching coding standards in programming," in *Proceedings of the 6th Conference on Information Technology Education*, ser. SIGITE '05. New York, NY, USA: Association for Computing Machinery, 2005, Conference Proceedings, pp. 239–244. [Online]. Available: <https://doi.org/10.1145/1095714.1095770>
- [162] C. Prasad, K. Sanders, and J. McTaggart, "Decoding programming concepts in EAL and EFL students: Is there a difference," in *17th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ)*, 2004, Conference Proceedings, pp. 168–173.
- [163] P. G. Feijóo-García, K. McNamara, and J. Stuart, "The effects of native language on block-based programming introduction: A work in progress with Hispanic population," in *Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, vol. 1, 2020, pp. 1–2. [Online]. Available: <https://doi.org/10.1109/RESPECT49803.2020.9272513>
- [164] S. Sayago and A. Bergantinos, "Exploring the first experiences of computer programming of older people with low levels of formal education: A participant observational case study," *International Journal of Human-Computer Studies*, vol.

- 148, p. 102577, 2021. [Online]. Available: <https://doi.org/10.1016/j.ijhcs.2020.102577>
- [165] Y. Lei and M. Allen, "English language learners in computer science education: A scoping review," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE 2022, vol. 1. New York, NY, USA: Association for Computing Machinery, 2022, Conference Proceedings, pp. 57–63. [Online]. Available: <https://doi.org/10.1145/3478431.3499299>
- [166] A. Zumor and A. Qasem, "Challenges of using EMI in teaching and learning of university scientific disciplines: Student voice." *International Journal of Language Education*, vol. 3, no. 1, pp. 74–90, 2019. [Online]. Available: <https://doi.org/10.26858/ijole.v1i1.7510>
- [167] S. Raffaa and N. L. Ali, "Attitudes and challenges in the implementation of EMI among saudi students," *International Journal of Academic Research in Business and Social Sciences*, vol. 13, no. 5, 2023.
- [168] W. W. Lau and A. H. Yuen, "The impact of the medium of instruction: The case of teaching and learning of computer programming," *Education and Information Technologies*, vol. 16, no. 2, pp. 183–201, 2011. [Online]. Available: <https://doi.org/10.1007/s10639-009-9118-8>
- [169] Y. Pal and S. Iyer, "Comparison of English Versus Hindi medium students for programming abilities acquired through video-based instruction," in *IEEE Fourth International Conference on Technology for Education*, 2012, pp. 26–30. [Online]. Available: <https://doi.org/10.1109/T4E.2012.30>
- [170] —, "Effect of medium of instruction on programming ability acquired through screencast," in *International Conference on Learning and Teaching in Computing and Engineering*, 2015, pp. 17–21. [Online]. Available: <https://doi.org/10.1109/LaTiCE.2015.38>
- [171] A. G. S. Raj, K. Ketsuriyonk, J. M. Patel, and R. Halverson, "What do students feel about learning programming using both English and their native language?" in *International Conference on Learning and Teaching in Computing and Engineering (LaTiCE)*. IEEE, 2017, Conference Proceedings, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/LaTiCE.2017.8>
- [172] A. G. Soosai Raj, K. Ketsuriyonk, J. M. Patel, and R. Halverson, "Does native language play a role in learning a programming language?" in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, Conference Proceedings, pp. 417–422. [Online]. Available: <https://doi.org/10.1145/3159450.3159531>
- [173] A. G. Soosai Raj, E. Zhang, S. Mukherjee, J. Williams, R. Halverson, and J. M. Patel, "Effect of native language on student learning and classroom interaction in an Operating Systems course," in *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '19. New York, NY,

- USA: Association for Computing Machinery, 2019, Conference Proceedings, pp. 499–505. [Online]. Available: <https://doi.org/10.1145/3304221.3319787>
- [174] A. G. S. Raj, H. Zhang, V. Abhyankar, S. Mukerjee, E. Zhang, J. Williams, R. Halverson, and J. M. Patel, “Impact of bilingual CS education on student learning and engagement in a Data Structures course,” in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '19. New York, NY, USA: Association for Computing Machinery, 2019, Conference Proceedings. [Online]. Available: <https://doi.org/10.1145/3364510.3364518>
- [175] I. Villegas Molina, A. Salguero, S. Zhong, and A. G. Soosai Raj, “The effects of Spanish-English bilingual instruction in a CS0 course for high school students,” in *Proceedings of the Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE 2023, vol. 1. New York, NY, USA: Association for Computing Machinery, 2023, pp. 75–81. [Online]. Available: <https://doi.org/10.1145/3587102.3588845>
- [176] S. Grover, A. Raman, and H. Banati, “Does bilingual specification impact students’ comprehension of problems in introductory programming?” in *Proceedings of the 16th Annual ACM India Compute Conference*, ser. COMPUTE '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 66â71. [Online]. Available: <https://doi-org.ucd.idm.oclc.org/10.1145/3627217.3627237>
- [177] S. D. White, S. May, and M. Washburn, “A study of freshmen,” *The American Journal of Psychology*, vol. 28, no. 1, pp. 151–155, 1917. [Online]. Available: <https://doi.org/10.2307/1412944>
- [178] W. McNamara and J. Hughes, “A review of research on the selection of computer programmers.” *Personnel Psychology*, pp. 39–51, 1961. [Online]. Available: <https://doi.org/10.1111/j.1744-6570.1961.tb00920.x>
- [179] A. Hellas, P. Ithantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom, and S. N. Liao, “Predicting academic performance: A systematic literature review,” in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '18 Companion. New York, NY, USA: Association for Computing Machinery, 2018, pp. 175–199. [Online]. Available: <https://doi.org/10.1145/3293881.3295783>
- [180] A. De Nicola and G. D’Agostino, “Assessment of gender divide in scientific communities,” *Scientometrics*, vol. 126, no. 5, pp. 3807–3840, 2021. [Online]. Available: <https://doi.org/10.1007/s11192-021-03885-3>
- [181] A. Lishinski, A. Yadav, J. Good, and R. Enbody, “Learning to program: Gender differences and interactive effects of students’ motivation, goals, and self-efficacy on performance,” in *Proceedings of the ACM Conference on International Computing Education Research*, ser. ICER '16. New York, NY, USA: Association for Computing Machinery, 2016, Conference Proceedings, pp. 211–220. [Online]. Available: <https://doi.org/10.1145/2960310.2960329>

- [182] M. F. M. Mohsin, M. H. A. Wahab, M. F. Zaiyadi, N. M. Norwawi, and C. F. Hibadullah, "An investigation into influence factor of student programming grade using association rule mining," vol. 2, no. 2, pp. 19–27, 2010. [Online]. Available: <https://repo.uum.edu.my/id/eprint/14738>
- [183] A. M. Shahiri, W. Husain, and N. A. Rashid, "A review on predicting student's performance using data mining techniques," *Procedia Computer Science*, vol. 72, pp. 414–422, 2015, (Peer-review under responsibility of organizing committee of Information Systems International Conference (ISICO2015)). [Online]. Available: <https://doi.org/10.1016/j.procs.2015.12.157>
- [184] M. A. Robinson, "Using multi-item psychometric scales for research and practice in human resource management," *Human Resource Management*, vol. 57, no. 3, pp. 739–750, 2018. [Online]. Available: <https://doi.org/10.1002/hrm.21852>
- [185] E. Botes, J.-M. Dewaele, and S. Greiff, "The foreign language classroom anxiety scale and academic achievement: an overview of the prevailing literature and a meta-analysis," *Journal for the Psychology of Language Learning*, vol. 2, no. 1, pp. 26–56, 2020. [Online]. Available: <https://jpll.org/index.php/journal/article/view/botesetal>
- [186] M. M. T. Rodrigo, R. S. Baker, M. C. Jadud, A. C. M. Amarra, T. Dy, M. B. V. Espejo-Lahoz, S. A. L. Lim, S. A. Pascua, J. O. Sugay, and E. S. Tabanao, "Affective and behavioral predictors of novice programmer achievement," in *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 156–160. [Online]. Available: <https://doi.org/10.1145/1562877.1562929>
- [187] F. Ofori, E. Maina, and R. Gitonga, "Using machine learning algorithms to predict students' performance and improve learning outcome: A literature based review," *Journal of Information and Technology*, vol. 4, no. 1, pp. 33–55, 2020.
- [188] A. Abu Saa, M. Al-Emran, and K. Shaalan, "Factors affecting students' performance in higher education: a systematic review of predictive data mining techniques," *Technology, Knowledge and Learning*, vol. 24, pp. 567–598, 2019. [Online]. Available: <https://doi.org/10.1007/s10758-019-09408-7>
- [189] M. Allen, C. Berg, J. Q. Dawson, and N. Leveridge, "Insights from the application of universal design principles to support English language learners," in *Proceedings of the 23rd Western Canadian Conference on Computing Education*, ser. WCCCE '18. New York, NY, USA: Association for Computing Machinery, 2018, Conference Proceedings. [Online]. Available: <https://doi.org/10.1145/3209635.3209646>
- [190] C. Piech, L. Yan, L. Einstein, A. Saavedra, B. Bozkurt, E. Sestakova, O. Guth, and N. McKeown, "Co-teaching computer science across borders: Human-centric learning at scale," in *Proceedings of the Seventh ACM Conference on Learning @ Scale*, ser. L@S '20. New York, NY, USA: Association for Computing

- Machinery, 2020, Conference Proceedings, pp. 103–113. [Online]. Available: <https://doi.org/10.1145/3386527.3405915>
- [191] M. Mamman, R. Shuaybu Rdadi, and M. Abubakar Sadiq, “Computerized E-learning platform for teaching Python programming using Hausa language,” *World Journal of Advanced Engineering Technology and Sciences*, vol. 02, pp. 116–123, 2023. [Online]. Available: <https://doi.org/10.30574/wjaets.2023.9.2.0214>
- [192] P. Doyle, M. Szappanyos, B. Gillespie, C. Ennis, V. Rebay, and R. B. Shapiro, “Flipped learning in computer science for non-native English-speaking students: Case study,” *Irish Journal of Academic Practice*, vol. 10, no. 1, 2022. [Online]. Available: <https://doi.org/10.21427/dnrm-d680>
- [193] Q. Xiong, J. Peng, and Z. Yang, “Study on bilingual teaching of computer science in Chinese universities,” in *International Conference on Educational and Information Technology*, vol. 2, 2010, pp. V2–226–V2–229. [Online]. Available: <https://doi.org/10.1109/ICEIT.2010.5607559>
- [194] H. Li, K. Wang, and Y. Wang, “A bilingual teaching modal in a programming language course,” in *the Second International Conference on Education Technology and Training*, 2009, pp. 197–200. [Online]. Available: <https://doi.org/10.1109/ETT.2009.71>
- [195] G. Jiang and C. Zhao, “Practice and exploration on bilingual teaching for Java programming language,” in *the International Conference on Educational and Information Technology*, vol. 2, 2010, pp. V2–465–V2–468. [Online]. Available: <https://doi.org/10.1109/ICEIT.2010.5607617>
- [196] S. Hagiwara and N. J. Rodriguez, “English learners (EL) and computer science (CS) learning: Equity issues,” in *Handbook of Research on Equity in Computer Science in P-16 Education*. IGI Global, 2021, pp. 70–87. [Online]. Available: <https://doi.org/10.4018/978-1-7998-4739-7.ch005>
- [197] Z. Liang and P. Ian, “An experience report on English Medium Instruction (EMI) based computing education in a faculty of Engineering in Japan,” in *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEET)*, 2023, pp. 163–167.
- [198] H. H. A. Razaq, A. S. Gaser, M. A. Mohammed, E. T. Yassen, S. A. Mostafad, S. R. Zeebaree, D. A. Ibrahim, M. K. Abd Ghania, and R. N. Farhan, “Designing and implementing an Arabic programming language for teaching pupils,” *Journal of Southwest Jiaotong University*, vol. 54, no. 3, 2019. [Online]. Available: <https://doi.org/10.35741/issn.0258-2724.54.3.11>
- [199] A. Swidan and F. Hermans, “A framework for the localization of programming languages,” in *Proceedings of the 2023 ACM SIGPLAN International Symposium on SPLASH-E*, ser. SPLASH-E 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 13–25. [Online]. Available: <https://doi.org/10.1145/3622780.3623645>

- [200] C. Piech and S. Abu-El-Haija, "Human languages in source code: Auto-translation for localized instruction," in *Proceedings of the Seventh ACM Conference on Learning @ Scale*, ser. L@S '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 167–174. [Online]. Available: <https://doi.org/10.1145/3386527.3405916>
- [201] P. Perera and S. Ahangama, "Simplytrans: A simplified approach to sinhala-based coding and introductory programming language localization," in *IEEE 16th International Conference on Industrial and Information Systems (ICIIS)*, 2021, pp. 318–323. [Online]. Available: <https://doi.org/10.1109/ICIIS53135.2021.9660709>
- [202] B. Roehner, "Translation into any natural language of the error messages generated by any computer program," *arXiv Preprint*, 2015. [Online]. Available: <https://doi.org/10.48550/arXiv.1508.04936>
- [203] C. Dunn and K. Ghosh, "BicePy: Bilingual description of compiler errors in Python," in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2023, pp. 229–231.
- [204] J.-Y. Shin, L. Q. Dixon, and Y. Choi, "An updated review on use of L1 in foreign language classrooms," *Journal of Multilingual and Multicultural Development*, vol. 41, no. 5, pp. 406–419, 2020. [Online]. Available: <https://doi.org/10.1080/01434632.2019.1684928>
- [205] R. F. Martínez-Valencia, J. A. Montero-Valverde, M. Martínez-Arroyo, and J. M. Hernández-Bravo, "A web app for teaching specialized English vocabulary - case of study: Computer sciences," in *Technologies and Innovation: 8th International Conference, CITI '22*, Guayaquil, Ecuador, November 2022, pp. 260–274. [Online]. Available: https://doi.org/10.1007/978-3-031-19961-5_19
- [206] J. Bi, "How large a vocabulary do Chinese computer science undergraduates need to read English-medium specialist textbooks?" *English for Specific Purposes*, vol. 58, pp. 77–89, 2020. [Online]. Available: <https://doi.org/10.1016/j.esp.2020.01.001>
- [207] B. Kaldarova, B. Omarov, L. Zhaidakbayeva, A. Tursynbayev, G. Beissenova, B. Kurmanbayev, and A. Anarbayev, "Applying game-based learning to a primary school class in computer science terminology learning," in *Frontiers in Education*, vol. 8, 2023, p. 1100275. [Online]. Available: <https://doi.org/10.3389/feduc.2023.1100275>
- [208] J. P. Da Silva and I. F. Silveira, "An open educational game based on re-purpose for teaching technical English in the context of computer programming at Brazil," vol. 3099, 2020. [Online]. Available: <https://ceur-ws.org/Vol-3099/paper17.pdf>
- [209] R. R. Simanjutak, "Learning specific academic vocabulary using MALL: experience from computer science students," *Teaching English with Technology*, vol. 20, no. 5, pp. 87–107, 2020. [Online]. Available: <https://www.ceeol.com/search/article-detail?id=912729>

- [210] D. S. Wahyuni, Y. Rozimela, H. Ardi, M. Mukhaiyar, and T. S. E. Fatdha, "Proposing a web-based interactive module for education for sustainable development in English for computer science," *Register Journal*, vol. 16, no. 1, pp. 1–23, 2023. [Online]. Available: <https://doi.org/10.18326/register.v16i1.1-23>
- [211] D. Davies, D. Jindal-Snape, C. Collier, R. Digby, P. Hay, and A. Howe, "Creative learning environments in education - a systematic literature review," *Thinking Skills and Creativity*, vol. 8, pp. 80–91, 2013. [Online]. Available: <https://doi.org/10.1016/j.tsc.2012.07.004>
- [212] R. Chambers, "Design considerations for introductory programming instruction," in *SITE Interactive Conference*. Association for the Advancement of Computing in Education (AACE), 2020, pp. 626–631. [Online]. Available: <https://www.learntechlib.org/primary/p/218213/>
- [213] A. Osanloo and C. Grant, "Understanding, selecting, and integrating a theoretical framework in dissertation research: Creating the blueprint for your "house"," *Administrative issues journal: connecting education, practice, and research*, vol. 4, no. 2, pp. 12–26, 2016. [Online]. Available: <https://dc.swosu.edu/aij/vol4/iss2/4>
- [214] D. M. Mertens, *Research and evaluation in education and psychology: Integrating diversity with quantitative, qualitative, and mixed methods*. Sage publications, 2019.
- [215] E. Guba, *The paradigm dialog*. Newbury, California: Sage Publications, Inc, 1990.
- [216] N. Mackenzie and S. Kriepke, "Research dilemmas: Paradigms, methods and methodology," *Issues in educational research*, vol. 16, no. 2, pp. 193–205, 2006. [Online]. Available: <http://www.iier.org.au/iier16/mackenzie.html>
- [217] L. A. Hudson and J. L. Ozanne, "Alternative ways of seeking knowledge in consumer research," *Journal of consumer research*, vol. 14, no. 4, pp. 508–521, 1988. [Online]. Available: <https://doi.org/10.1086/209132>
- [218] J. Scotland, "Exploring the philosophical underpinnings of research: Relating ontology and epistemology to the methodology and methods of the scientific, interpretive, and critical research paradigms," *English language teaching*, vol. 5, no. 9, pp. 9–16, 2012. [Online]. Available: <http://dx.doi.org/10.5539/elt.v5n9p9>
- [219] M. J. Crotty, "The foundations of social research: Meaning and perspective in the research process," *The foundations of social research*, pp. 1–256, 1998.
- [220] Z. O'Leary, *The essential guide to doing research*. SAGE Publications Ltd, 2004.
- [221] L. Malmi, J. Sheard, Simon, R. Bednarik, J. Helminen, P. Kinnunen, A. Korhonen, N. Myller, J. Sorva, and A. Taherkhani, "Theoretical underpinnings of computing education research: What is the evidence?" in *Proceedings of the Tenth Annual Conference on International Computing Education Research*, ser. ICER '14. New York, NY, USA: Association for Computing Machinery, 2014, Conference Proceedings, pp. 27–34. [Online]. Available: <https://doi.org/10.1145/2632320.2632358>

- [222] S. A. Fincher and A. V. Robins, *The Cambridge handbook of computing education research*. Cambridge University Press, 2019.
- [223] A. H. Panhwar, S. Ansari, and A. A. Shah, "Post-positivism: An effective paradigm for social and educational research," *International Research Journal of Arts Humanities (IRJAH)*, vol. 45, no. 45, pp. 253–259, 2017.
- [224] D. C. Phillips and N. C. Burbules, *Postpositivism and educational research*. Rowman & Littlefield, 2000.
- [225] A. B. Ryan, *Researching and Writing your Thesis: a guide for postgraduate students*. MACE: Maynooth Adult and Community Education, 2006, ch. Post-positivist approaches to research, pp. 12–26. [Online]. Available: <https://eprints.maynoothuniversity.ie/874/>
- [226] R. E. Stake, *The art of case study research*. SAGE Publications, Inc, 1995.
- [227] P. D. Leedy and J. E. Ormrod, *Practical research: Planning and Design*, 11th ed. Essex, England: Pearson Education, 2016.
- [228] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, Quantitative, and Mixed Methods Approaches*, 3rd ed. SAGE Publications, Inc., 2018.
- [229] J. W. Cresswell, *Qualitative inquiry and research design: Choosing among five traditions*. Thousand Oaks, California, US: Sage Publications, Inc., 1998.
- [230] L. Luck, D. Jackson, and K. Usher, "Case study: a bridge across the paradigms," *Nursing inquiry*, vol. 13, no. 2, pp. 103–109, 2006. [Online]. Available: <https://doi.org/10.1111/j.1440-1800.2006.00309.x>
- [231] M. Lewis-Beck, A. E. Bryman, and T. F. Liao, *The Sage encyclopedia of social science research methods*. Sage Publications, 2003.
- [232] R. K. Yin, *Case Study Research and Applications: Design and Methods*, 6th ed. SAGE Publications, Inc, 2018, vol. 5.
- [233] A. Erickson, "Case Studies," in *The Students' Guide to Learning Design and Research*, R. Kimmons and S. Caskurlu, Eds. EdTech books, 2020. [Online]. Available: <https://dx.doi.org/10.59668/10>
- [234] H. K. Pae, *Writing systems, reading processes, and cross-linguistic influences: Reflections from the Chinese, Japanese and Korean languages*. John Benjamins Publishing Company, 2018, vol. 7.
- [235] K. Versteegh, "Linguistic contacts between Arabic and other languages," *Arabica*, vol. 48, no. 4, pp. 470–508, 2001. [Online]. Available: <http://www.jstor.org/stable/4057668>
- [236] W. Foddy and W. H. Foddy, *Constructing questions for interviews and questionnaires: Theory and practice in social research*. Cambridge university press, 1993.

- [237] R. Reja, K. L. Manfreda, V. Hlebec, and V. Vehovar, "Open-ended vs. close-ended questions in web questionnaires," *Developments in applied statistics*, vol. 19, no. 1, pp. 159–177, 2003.
- [238] Education-First. (2020) EF Standard English Test. [Online]. Available: <https://www.efset.org>
- [239] ——. (2020) EF Standard English Test. [Online]. Available: <https://www.efset.org/english-score/>
- [240] M. C. Rodriguez, "Three options are optimal for multiple-choice items: A meta-analysis of 80 years of research," *Educational measurement: issues and practice*, vol. 24, no. 2, pp. 3–13, 2005. [Online]. Available: <https://doi.org/10.1111/j.1745-3992.2005.00006.x>
- [241] J. Schroeder, K. L. Murphy, and T. A. Holme, "Investigating factors that influence item performance on ACS exams," *Journal of Chemical Education*, vol. 89, no. 3, pp. 346–350, 2012. [Online]. Available: <https://doi.org/10.1021/ed101175f>
- [242] Y. Aida, "Examination of Horwitz, Horwitz, and Cope's construct of foreign language anxiety: The case of students of Japanese," *The modern language journal*, vol. 78, no. 2, pp. 155–168, 1994. [Online]. Available: <https://doi.org/10.2307/329005>
- [243] S. Matsuda and P. Gobel, "Anxiety and predictors of performance in the foreign language classroom," *System*, vol. 32, no. 1, pp. 21–36, 2004. [Online]. Available: <https://doi.org/10.1016/j.system.2003.08.002>
- [244] H. Elkhafaifi, "Listening comprehension and anxiety in the Arabic language classroom," *The modern language journal*, vol. 89, no. 2, pp. 206–220, 2005. [Online]. Available: <https://doi.org/10.1111/j.1540-4781.2005.00275.x>
- [245] M. Toyama and Y. Yamazaki, "Exploring the components of the foreign language classroom anxiety scale in the context of Japanese undergraduates," *Asian-Pacific Journal of Second and Foreign Language Education*, vol. 3, no. 1, pp. 1–27, 2018. [Online]. Available: <https://doi.org/10.1186/s40862-018-0045-3>
- [246] Y. Cheng, E. K. Horwitz, and D. L. Schallert, "Language anxiety: Differentiating writing and speaking components," *Language learning*, vol. 49, no. 3, pp. 417–446, 1999. [Online]. Available: <https://doi.org/10.1111/0023-8333.00095>
- [247] M. Liu and J. Jackson, "An exploration of Chinese EFL learners' unwillingness to communicate and foreign language anxiety," *The Modern Language Journal*, vol. 92, no. 1, pp. 71–86, 2008. [Online]. Available: <https://doi.org/10.1111/j.1540-4781.2008.00687.x>
- [248] T. Yashima, N. Kimberly A, T. Shizuka, O. Takeuchi, S. Yamane, and K. Yoshizawa, "The interplay of classroom anxiety, intrinsic motivation, and gender in the Japanese EFL context," *Journal of Foreign Language Education and Research*, vol. 17, no. 3, pp. 41–64, 2009. [Online]. Available: https://www.kansai-u.ac.jp/fl/publication/pdf_education/17/00_yashima_44.pdf

- [249] B. Mak, "An exploration of speaking-in-class anxiety with chinese ESL learners," *System*, vol. 39, no. 2, pp. 202–214, 2011. [Online]. Available: <https://doi.org/10.1016/j.system.2011.04.002>
- [250] G. Park, "Factor analysis of the foreign language classroom anxiety scale in Korean learners of English as a foreign language," *Psychological Reports*, vol. 115, no. 1, pp. 261–275, 2014. [Online]. Available: <https://doi.org/10.2466/28.11.PR0.115c10z2>
- [251] S. Alaofi and S. Russell, "The influence of foreign language classroom anxiety on academic performance in English-based CS1 courses," in *Proceedings of the Conference on United Kingdom & Ireland Computing Education Research*, ser. UKICER '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://doi.org/10.1145/3555009.3555020>
- [252] M. Tavakol and R. Dennick, "Making sense of Cronbach's alpha," *International Journal of Medical Education*, vol. 2, pp. 53–55, 2011. [Online]. Available: <https://doi.org/10.5116/ijme.4dfb.8dfd>
- [253] M. Azher, M. N. Anwar, and A. Naz, "An investigation of foreign language classroom anxiety and its relationship with students achievement," *Journal of College Teaching Learning (TLC)*, vol. 7, no. 11, 2010. [Online]. Available: <https://doi.org/10.19030/tlc.v7i11.249>
- [254] S. Alaofi and S. Russell, "Computer terminology test for non-native English speaking CS1 students," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, Conference Proceedings, pp. 1304–1304. [Online]. Available: <https://doi.org/10.1145/3408877.3439593>
- [255] —, "A validated computer terminology test for predicting non-native English-speaking CS1 students' academic performance," in *Proceedings of the 24th Australasian Computing Education Conference*, ser. ACE '22. New York, NY, USA: Association for Computing Machinery, 2022, Conference Proceedings, pp. 133–142. [Online]. Available: <https://doi.org/10.1145/3511861.3511876>
- [256] IBM, "SPSS Statistics," 2019.
- [257] N. J. Gogtay and U. M. Thatte, "Principles of correlation analysis," *Journal of the Association of Physicians of India*, vol. 65, no. 3, pp. 78–81, 2017. [Online]. Available: https://www.kem.edu/wp-content/uploads/2012/06/9-Principles_of_correlation-1.pdf
- [258] M. Tranmer, J. Murphy, M. Elliot, and M. Pampaka, *Multiple linear regression*, 2nd ed. The Cathie Marsh Centre for Census and Survey Research (CCSR), 2020. [Online]. Available: <https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/multiple-linear-regression.pdf>
- [259] H. Kyngäs, *Inductive Content Analysis*. Cham: Springer International Publishing, 2020, pp. 13–21. [Online]. Available: https://doi.org/10.1007/978-3-030-30199-6_2

- [260] S. B. Merriam, *Qualitative Research and Case Study Applications in Education. Revised and Expanded from "Case Study Research in Education"*. ERIC, 1998.
- [261] J. Barnes, K. Conrad, C. Demont-Heinrich, M. Graziano, D. Kowalski, J. Neufeld, J. Zamora, and M. Palmquist. (2005) Generalizability and Transferability. [Online]. Available: <https://writing.colostate.edu/guides/guide.cfm?guideid=65>
- [262] R. Wicklin. (2017, September) Fisher's transformation of the correlation coefficient. SAS Analytics Software Solutions. (Last accessed January 05, 2023). [Online]. Available: <https://blogs.sas.com/content/iml/2017/09/20/fishers-transformation-correlation.html>
- [263] K. A. McNeil, I. Newman, and F. J. Kelly, *Testing research hypotheses with the general linear model*. Southern Illinois University Press, 1996.
- [264] K. Quille and S. Bergin, "CS1: how will they do? how can we help? a decade of research and practice," *Computer Science Education*, vol. 29, no. 2-3, pp. 254–282, 2019. [Online]. Available: <https://doi.org/10.1080/08993408.2019.1612679>
- [265] J. T. Roscoe, *Fundamental research statistics for the behavioral sciences*, 1975.
- [266] R. M. Conroy, "The RCSI sample size handbook," 2021. [Online]. Available: <https://www.beaumontethics.ie/docs/application/samplesize2021.pdf>
- [267] T. Walsh, P. Nurkka, H. Petrie, and J. Olsson, "The effect of language in answering qualitative questions in user experience evaluation web-surveys," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, ser. OzCHI '13. New York, NY, USA: Association for Computing Machinery, 2013, Conference Proceedings, pp. 73–82. [Online]. Available: <https://doi.org/10.1145/2541016.2541049>
- [268] A. Coxhead, "A new academic word list," *TESOL Quarterly*, vol. 34, no. 2, pp. 213–238, 2000. [Online]. Available: <https://doi.org/10.2307/3587951>
- [269] J. Wang, S. Liang, and G. Ge, "Establishment of a medical academic word list," *English for Specific Purposes*, vol. 27, no. 4, pp. 442–458, 2008. [Online]. Available: <https://doi.org/10.1016/j.esp.2008.05.003>
- [270] J. Ward, "A basic engineering English word list for less proficient foundation engineering undergraduates," *English for Specific Purposes*, vol. 28, no. 3, pp. 170–182, 2009. [Online]. Available: <https://doi.org/10.1016/j.esp.2009.04.001>
- [271] N. Konstantakis, "Creating a business word list for teaching business English," *ELIA*, vol. 7, pp. 79–102, 2007. [Online]. Available: <http://hdl.handle.net/11441/34157>
- [272] D. Roesler, "A computer science academic vocabulary list," Masters dissertation, Portland State University, 2020. [Online]. Available: <https://doi.org/10.15760/etd.7414>

- [273] S. Y. Uba, J. Irudayasamy, and C. A. Hankins, "A new computer science academic word list," *World Journal of English Language*, vol. 13, no. 5, 2023. [Online]. Available: <https://doi.org/10.5430/wjel.v13n5p319>
- [274] L. Anthony, "Antfileconverter," Waseda University, Tokyo, Japan, 2017, (version 1.2.1). [Online]. Available: <https://www.laurenceanthony.net/software/antfileconverter/>
- [275] —, "Antwordprofiler," Waseda University, Tokyo, Japan, 2013, (version 1.4.0.1). [Online]. Available: <https://www.laurenceanthony.net/software/antwordprofiler/>
- [276] —, "Antconc," Waseda University, Tokyo, Japan, 2019, (version 3.5.8). [Online]. Available: <https://www.laurenceanthony.net/software/antconc/>
- [277] I. S. P. Nation, *Making and using word lists for language learning and testing*. John Benjamins Publishing Company, 2016.
- [278] B. A. Becker, D. Gallagher, P. Denny, J. Prather, C. Gostomski, K. Norris, and G. Powell, "From the horse's mouth: The words we use to teach diverse student groups across three continents," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE 2022, vol. 1. New York, NY, USA: Association for Computing Machinery, 2022, pp. 71–77. [Online]. Available: <https://doi.org/10.1145/3478431.3499392>
- [279] R. F. Burton, "Multiple-choice and true/false tests: myths and misapprehensions," *Assessment & Evaluation in Higher Education*, vol. 30, no. 1, pp. 65–72, 2005. [Online]. Available: <https://doi.org/10.1080/0260293042003243904>
- [280] M. G. Jodoin, A. Zenisky, and R. K. Hambleton, "Comparison of the psychometric properties of several computer-based test designs for credentialing exams with multiple purposes," *Applied Measurement in Education*, vol. 19, no. 3, pp. 203–220, 2006. [Online]. Available: https://doi.org/10.1207/s15324818ame1903_3
- [281] S. M. Downing and T. M. Haladyna, "Validity threats: overcoming interference with proposed interpretations of assessment data," *Medical Education*, vol. 38, pp. 327–333, 2004. [Online]. Available: <https://doi.org/10.1046/j.1365-2923.2004.01777.x>
- [282] Computer hope: Free computer terms, dictionary, and glossary. (Last accessed March 13, 2021). [Online]. Available: www.computerhope.com/jargon.htm
- [283] Techterms: The computer dictionary. (Last accessed March 13, 2021). [Online]. Available: techterms.com
- [284] R. Vyas and A. Supe, "Multiple choice questions: a literature review on the optimal number of options," *The National Medical Journal of India*, vol. 21, no. 3, pp. 130–3, 2008.

- [285] T. Brothen, "Time limits on tests: Updating the 1-minute rule," *Teaching of Psychology*, vol. 39, no. 4, pp. 288–292, 2012. [Online]. Available: <https://doi.org/10.1177/0098628312456630>
- [286] T. L. Kelley, "Interpretation of educational measurements," 1927.
- [287] S. Messick, *Validity*, 3rd ed. New York: Macmillan, 1989, pp. 13–103.
- [288] —, "Validity and washback in language testing," *Language testing*, vol. 13, no. 3, pp. 241–256, 1996. [Online]. Available: <https://doi.org/10.1177/02655322960130030>
- [289] E. V. Smith Jr., "Evidence for the reliability of measures and validity of measure interpretation: a Rasch measurement perspective," *Journal of applied measurement*, 2001. [Online]. Available: <http://europepmc.org/abstract/MED/12011511>
- [290] G. Rasch, "Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests," 1960.
- [291] P. Susongko, "Validation of science achievement test with the Rasch model," *Jurnal Pendidikan IPA Indonesia*, vol. 5, no. 2, pp. 268–277, 2016. [Online]. Available: <https://journal.unnes.ac.id/nju/index.php/jpii/article/view/7690>
- [292] E. James Sick, "Rasch measurement in language education part 4: Rasch analysis software programs," *JALT Testing Evaluation SIG Newsletter*, vol. 13, no. 3, pp. 13–16, November 2009. [Online]. Available: https://hosted.jalt.org/test/sic_4.htm
- [293] T. Bond, "Validity and assessment: a Rasch measurement perspective," *Metodología de las Ciencias del Comportamiento*, vol. 5, pp. 179–194, 2004. [Online]. Available: <https://researchonline.jcu.edu.au/1799/>
- [294] E. W. Wolfe and E. V. Smith Jr, "Instrument development tools and activities for measure validation using Rasch models: part ii- validation activities," *Journal of applied measurement*, vol. 8, no. 2, pp. 204–234, 2007. [Online]. Available: <http://europepmc.org/abstract/MED/17440262>
- [295] P. Baghaei and N. Amrahi, "Validation of a multiple choice English vocabulary test with the Rasch model," *Journal of Language Teaching and Research*, vol. 2, no. 5, pp. 1052–1060, 2011. [Online]. Available: <https://doi.org/10.4304/jltr.2.5.1052-1060>
- [296] P. Baghaei, "The Rasch model as a construct validation tool," *Rasch Measurement Transactions*, vol. 22, no. 1, pp. 1145–1146, 2008.
- [297] B. D. Wright and M. Stone, *Measurement essentials*, 2nd ed. Wilmington, Delaware: Wide Range, Inc, 1999.
- [298] J. Runnels, "Using the Rasch model to validate a multiple choice English achievement test," *International Journal of Language Studies*, vol. 6, no. 4, pp. 141–155, 2012.

- [299] V. Aryadoust, H. A. H. Tan, and L. Y. Ng, "A scientometric review of Rasch measurement: The rise and progress of a specialty," *Frontiers in psychology*, vol. 10, p. 2197, 2019. [Online]. Available: <https://doi.org/10.3389/fpsyg.2019.02197>
- [300] T. McNamara and U. Knoch, "The Rasch wars: The emergence of Rasch measurement in language testing," *Language Testing*, vol. 29, no. 4, pp. 555–576, 2012. [Online]. Available: <https://doi.org/10.1177/0265532211430367>
- [301] D. Beglar, "A Rasch-based validation of the vocabulary size test," *Language testing*, vol. 27, no. 1, pp. 101–118, 2010. [Online]. Available: <https://doi.org/10.1177/0265532209340194>
- [302] J. Runnels, "Evaluation of an achievement English vocabulary test using Rasch analysis," pp. 151–171, 2011. [Online]. Available: <http://id.nii.ac.jp/1092/00000937>
- [303] Z. M. Maskor, H. Baharudin, and M. A. Lubis, "Measurement validity and reliability of the productive vocabulary knowledge instrument for Arabic learners in Malaysian secondary schools," *Advanced Science Letters*, vol. 24, no. 5, pp. 3423–3426, 2018. [Online]. Available: <https://doi.org/10.1166/asl.2018.11400>
- [304] M. Linacre, "Winsteps," 2021. [Online]. Available: <http://www.winsteps.com>
- [305] —, *A User's Guide to WINSTEPS ®/MINISTEP Rasch-Model Computer Programs*. winsteps.com, 2021. [Online]. Available: <https://www.winsteps.com/a/Winsteps-Manual.pdf>
- [306] T. G. Bond and C. M. Fox, *Applying the Rasch model: Fundamental measurement in the human sciences*, 2nd ed. Mahwah, New Jersey: Lawrence Erlbaum Associates, 2007.
- [307] K. D. Royal, "Making meaningful measurement in survey research: The use of person and item maps," *Online Submission ERIC*, 2009.
- [308] W. J. Boone and K. Scantlebury, "The role of Rasch analysis when conducting science education research utilizing multiple-choice tests," *Science Education*, vol. 90, no. 2, pp. 253–269, 2006. [Online]. Available: <https://doi.org/10.1002/sce.20106>
- [309] R. T. Warne, *Statistics for the social sciences: a general linear model approach*, 2nd ed. Cambridge University Press, 2020. [Online]. Available: <https://doi.org/10.1017/9781108894319>
- [310] O. Chirobocea-Tudor, "Translation as a vocabulary teaching aid in ESP," *Buletinul Stiintific al Universitatii Politehnica din Timisoara, Seria Limbi Moderne*, vol. 20, pp. 76–87, 2021. [Online]. Available: <https://dspace.upt.ro/xmlui/handle/123456789/5736>

- [311] M. Alhamami, "Inequity, inequality, and language rights in English as a medium of instruction programs," *Evaluation and Program Planning*, vol. 99, p. 102297, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0149718923000745>
- [312] Y. Zhan, Z. H. Wan, and D. Sun, "Online formative peer feedback in Chinese contexts at the tertiary level: A critical review on its design, impacts and influencing factors," *Computers Education*, vol. 176, p. 104341, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360131521002189>
- [313] M. Van Zundert, D. Sluijsmans, and J. Van Merriënboer, "Effective peer assessment processes: Research findings and future directions," *Learning and Instruction*, vol. 20, no. 4, pp. 270–279, 2010, unravelling Peer Assessment. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959475209000814>
- [314] E. Van Popta, M. Kral, G. Camp, R. L. Martens, and P. R. Simons, "Exploring the value of peer feedback in online learning for the provider," *Educational Research Review*, vol. 20, pp. 24–34, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1747938X16300562>
- [315] L. Hsia, I. Huang, and G. Hwang, "Effects of different online peer-feedback approaches on students' performance skills, motivation and self-efficacy in a dance course," *Computers & Education*, vol. 96, pp. 55–71, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360131516300240>
- [316] D. Nicol, A. Thomson, and C. Breslin, "Rethinking feedback practices in higher education: a peer review perspective," *Assessment & Evaluation in Higher Education*, vol. 39, no. 1, pp. 102–122, 2014. [Online]. Available: <https://doi.org/10.1080/02602938.2013.795518>
- [317] G. E. Kennedy and Q. I. Cutts, "The association between students' use of an electronic voting system and their learning outcomes," *Journal of Computer Assisted Learning*, vol. 21, no. 4, pp. 260–268, 2005. [Online]. Available: <https://doi.org/10.1111/j.1365-2729.2005.00133.x>
- [318] G. Lin, "Anonymous versus identified peer assessment via a Facebook-based learning application: Effects on quality of peer feedback, perceived learning, perceived fairness, and attitude toward the system," *Computers & Education*, vol. 116, pp. 81–92, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360131517301963>
- [319] A. Aldiab, H. Chowdhury, A. Kootsookos, F. Alam, and H. Allhibi, "Utilization of learning management systems (LMSs) in higher education system: A case review for saudi arabia," in *2nd International Conference on Energy and Power (ICEP)*, vol. 160, Sydney, Australia, 2019, pp. 731–737. [Online]. Available: <https://doi.org/10.1016/j.egypro.2019.02.186>

- [320] D. Turnbull, R. Chugh, and J. Luck, "Transitioning to e-learning during the COVID-19 pandemic: How have higher education institutions responded to the challenge?" *Education and Information Technologies*, vol. 26, pp. 6401–6419, 2021. [Online]. Available: <https://doi.org/10.1007/s10639-021-10633-w>
- [321] C. C. Aydin and G. Tirkes, "Open source learning management systems in e-learning and Moodle," in *IEEE EDUCON 2010 Conference*, Madrid, Spain, 2010, pp. 593–600. [Online]. Available: <https://doi.org/10.1109/EDUCON.2010.5492522>
- [322] P. Kiyenia. (2023) What students hate about their LMS. (Last accessed May 23, 2023). [Online]. Available: <https://elearningindustry.com/what-students-hate-about-their-lms>
- [323] G. Henrick, "Moodle as a the central hub of learning with tools plugged in - learning tool interoperability," in *1st Moodle Research Conference (MRC2012)*, 2012, pp. 208–2111. [Online]. Available: <http://research.moodle.org/39/>
- [324] H. Manzoor, "Disseminating learning tools interoperability standards," Masters dissertation, Virginia Polytechnic Institute and State University, 2019. [Online]. Available: <http://hdl.handle.net/10919/90772>
- [325] R. S. Jhangiani, I. A. Chiang, C. Cuttler, and D. C. Leighton, *Research methods in psychology*, 4th ed. Kwantlen Polytechnic University, 2019.
- [326] D. T. Campbell and T. D. Cook, "Quasi-experimentation," *Chicago, IL: Rand McNally*, 1979.
- [327] M. Gopalan, K. Rosinger, and J. B. Ahn, "Use of quasi-experimental research designs in education research: Growth, promise, and challenges," *Review of Research in Education*, vol. 44, no. 1, pp. 218–243, 2020. [Online]. Available: <https://doi.org/10.3102/0091732X20903302>
- [328] H. J. Seltman, *Experimental Design and Analysis*. Carnegie Mellon University, 2018. [Online]. Available: <https://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>
- [329] J. Frost, *Hypothesis testing: An intuitive guide for making data driven decisions*. Statistics By Jim Publishing, 2020.
- [330] Wellbeing@school. (2023) Understanding and interpreting box plots. (Last accessed May 23, 2023). [Online]. Available: <https://www.wellbeingatschool.org.nz/information-sheet/understanding-and-interpreting-box-plots>
- [331] M. T. Kaiser and S. Y. Chowdhury, "Foreign language virtual class room: Anxiety creator or healer?" *English Language Teaching*, vol. 13, no. 11, pp. 130–139, 2020. [Online]. Available: <https://doi.org/10.5539/elt.v13n11p130>
- [332] E. Botes, J.-M. Dewaele, and S. Greiff, "The foreign language classroom anxiety scale and academic achievement: An overview of the prevailing literature and a meta-analysis," *Journal for the Psychology of Language Learning*, vol. 2, no. 1, pp. 26–56, 2020. [Online]. Available: <https://jpll.org/index.php/journal/issue/view/2>

- [333] Y. Alghazo and H. Al-Wadi, "English language anxiety and its effect on students' mathematics achievement," *International Education Studies*, vol. 12, no. 12, pp. 130–140, 2019. [Online]. Available: <https://doi.org/10.5539/ies.v12n12p130>
- [334] A. A. Khuwaileh, "IT terminology and translation: Cultural, lexicographic and linguistic problems," *Language for special purposes Journal (LSP)*, vol. 1, no. 2, 2010. [Online]. Available: <https://rauli.cbs.dk/index.php/lspcog/article/view/3089>
- [335] W.-P. Wen and R. Clément, "A Chinese conceptualisation of willingness to communicate in ESL," *Language Culture and Curriculum*, vol. 16, no. 1, pp. 18–38, 2003. [Online]. Available: <https://doi.org/10.1080/07908310308666654>
- [336] S. Alrasheedi, "Investigation of factors influencing speaking performance of Saudi EFL learners," *Arab World English Journal (AWEJ)*, vol. 11, no. 4, 2020. [Online]. Available: <https://ssrn.com/abstract=3764220>

APPENDICES

Appendix A: Background Survey

Part One: Demographic Data

1. Age:

- 18-23
- 24-29
- over 30
- Prefer not to answer

2. Gender:

- Female
- Male
- Other
- Prefer not to answer

3. Native language

- Chinese (Mandarin, Cantonese or Hunanese)
- English
- Hindi
- Spanish
- French
- Arabic
- Other
- Please specify (.....)

4. **One or both of my parents are native English speakers.**

- Yes
- No

Part Two: Academic Background

5. **I studied high school in a school that uses an English medium education system. (School that uses the English language to teach all subjects).**

- Yes
- No

6. I have done three months or more English language course to study English as a foreign language in one of the English-speaking countries. For example: UK, Ireland, USA, Australia, New Zealand.

- Yes
- No

7. I have done the following subjects at high school: (you can select more than one)

- Math
- Science (Physics or Chemistry or both)
- English Language (as a foreign language)
- Computer Science or Computer Basics.

8. I have had a standardised English language test before e.g. IELTS, TOEFL, etc.:

- Yes
- No

9. I have had the following standardised English language test:

- IELTS.
- TOEFL.
- STEP.
- Gaokao.
- Other

- Please provide your score if possible, in the box below:

[-----]

10. If the test you have taken is not listed above, please provide the test name and your score if possible:

[-----]

Part Three: Previous Computer Programming Experience

11. This is my first time taking a formal programming class or course.

- Yes
- No

12. **I learned to program on my own through books, online materials or with the help of friends or family members, etc. but did not take any class or course.**

- Yes
- No

– If Yes, please specify the language/s you have learned in the text box below:

[.....]

13. **In general, how do you evaluate your level of programming prior to this module:**

- No experience (*never learned or used any programming languages*)
- Novice (*I have some knowledge of basic programming concepts, but I am not able to write or execute any programming code without help.*)
- Intermediate (*I am able to write and execute a simple programming code by myself but not able to correct the errors that appear in my code without help.*)
- Advanced (*I am able to write and execute more complicated programming code and correct most of the errors that appear in my code.*)
- Expert (*I am able to develop a complete piece of software or application using a programming language and correct all errors that appear in my code.*)

14. **Prior to this module, I took a programming course in the following programming language/s: (you can select more than one)**

- Block-based visual programming language e.g. Scratch, BYOB.
- Object-oriented programming language e.g. Java, C++.
- Procedural programming language e.g. PASCAL, BASIC, COBOL.
- Web design using JavaScript programming language.
- Web design using HTML programming language.
- Another type of programming language.

– If you select the last option, please specify programming language/s in the text box below:

[.....]

Part Four: Using the English language in the current programming class

15. **The language I use in my current programming class:**

- I can ask or answer questions during the programming lecture in my native language.

- I can NOT use my native language to ask or answer questions during the programming lecture in my native language.

16. The language my teacher uses in current programming class:

- My teacher uses only the English language to teach this module.
- My teacher usually uses English in class but sometimes uses the native language of the students to explain difficult topics.
- My teacher uses only the native language, NOT English, to teach this module.

17. The language used in exams and assignments in the current programming class:

- All exams and assignments for this module are in English and the answers should be in English too.
- All exams and assignments for this module are in English and the answers should be in English too, but the teacher usually explains the questions to students using their native language.
- All exams and assignment questions for this module are in English but my teacher sometimes accepts answers in my native language.

18. The language used in books and materials for my current programming class:

- All of the lecture notes and other learning materials provided by my teacher are in English language.
- Some of the lecture notes and other learning materials provided by my teacher are in my native language.

Part Five: Opinions about studying programming in English

19. Please rate how frequently you face the following difficulties stated in each of the following statements regarding the use of the English language in your programming course.

	Never	Rarely	Sometimes	Often	Always
1- I find it difficult to understand some English computer terminology used in the class.					

Continued on next page

2- I find it difficult to understand programming concepts when they are explained in English.					
3- I find it difficult to understand or follow up with the teacher when s/he speaks only English language during programming class.					
4- I find it difficult to ask or answer questions in English during the lectures.					
5- I find it difficult to read or understand lecture notes or other learning materials in English.					
6- I find it difficult to read or understand exam or assignment questions in English.					
7- I find it difficult to write or explain my answers to exams or assignment questions in English even if I know the answer.					
8- Sometimes I give the wrong answer because of a misunderstanding of questions in the English language.					

20. If you are facing other difficulties, please explain it in the text box below:

- [.....]

21. If I have the choice, I will choose to study programming through:

- My native language
- English language
- Both languages at the same time.

22. In my opinion, studying programming through English language make the programming course:

- Easier.
- More difficult.
- Has no impact on the level of the module difficulty.

23. Regarding my future study and career, I think studying programming through the English language is:

- Extremely useless

- Somewhat useless
- Neither useful nor useless
- Somewhat useful
- Extremely useful

24. To me, the use of the English language when learning computer programming is:

- Not a barrier
- Somewhat of a barrier
- Moderate barrier
- Extreme barrier

Part Six: Programming language difficulty

25. For each of the following statements choose the option that most accurately describes your experience with the programming language you are learning in your programming course e.g. (C, C++, Java, etc.):

	Very Diffi- cult	Difficult	Neutral/ Neither easy nor hard	Easy	Very Easy
1- Writing code using English letters/ alphabet.					
2- Remember the correct spelling of the programming language keywords when writing programming code.					
3- Reading programming code					
4- Reading error messages					
5- Understanding and using error mes- sages					
6- Intuitively understanding the pro- gramming language keywords mean- ing and functionality					

Continued on next page

7- Understanding the relationship between the programming language keywords and their meaning or functionality					
8- Using the English interface for the programming software that I use to write and execute the code (IDE).					
9- Understanding or using the IDE hints/tips and help commands in the English language.					

26. If you are facing other difficulties, please explain it in the text box below:

- [.....]

27. Regarding my future study and career, I think studying computer programming languages that are based on the English language are :

- Extremely useless
- Somewhat useless
- Neither useful nor useless
- Somewhat useful
- Extremely useful

----- End of the Survey¹-----

¹A simplified Chinese copy of this survey is available here, and the same survey in Arabic language is available here.

Appendix B: Computer Terminology Test

Please select the correct terminology that matches the definition in each of the following questions:

1. A number system that only uses two digits — 0 and 1
 - a. **binary**
 - b. octal
 - c. hexadecimal
2. The process by which a computer carries out the instructions of a program.
 - a. **execution**
 - b. processing
 - c. programming
3. A sequence of instructions that is continually repeated until a certain condition is reached.
 - a. **loop**
 - b. condition
 - c. class
4. The spelling and grammar of instructions in a programming language.
 - a. **syntax**
 - b. semantics
 - c. rules
5. Which of the following terms describes the type of value a variable has and the type of operations the variable can be used in:
 - a. **data type**
 - b. data structure
 - c. data classification
6. A single repetition of a process or set of instructions in a computer program.
 - a. **iteration**
 - b. initialization
 - c. processing
7. The process of locating and removing computer program errors.

- a. **debugging**
 - b. troubleshooting
 - c. testing
- 8. The process of assigning an initial value to a variable.
 - a. **initialization**
 - b. assignment
 - c. installation
- 9. Names that work like containers to hold numbers, phrases, or other important data used in several places in the code.
 - a. **variables**
 - b. functions
 - c. holders
- 10. Type of expressions used to compare values and give only a true or false result.
 - a. **boolean**
 - b. binary
 - c. pointer
- 11. The specific reserved words in a programming language.
 - a. **keywords**
 - b. variables
 - c. identifiers
- 12. The name that the user gives to variable, function or class.
 - a. **identifier**
 - b. given-name
 - c. forename
- 13. Commands that enable a program to take decisions about following one path or another:
 - a. **control structure**
 - b. control flow
 - c. control path
- 14. A whole number (not a fraction) that can be positive, negative, or zero:
 - a. **integer**

- b. decimal
 - c. float
15. A named section of the code that performs a specific task:
- a. **function**
 - b. parameters
 - c. package
16. Stop a running program:
- a. **exit**
 - b. execute
 - c. delete
17. A set of instructions designed to perform a specific task:
- a. **algorithm**
 - b. arithmetic
 - c. algebra
18. A punctuation mark used to separate elements:
- a. **comma**
 - b. bracket
 - c. command
19. A variable that is used to keep track of anything that must be counted:
- a. **counter**
 - b. tracker
 - c. connector
20. The process of adding a value to a numerical value:
- a. **increment**
 - b. decrement
 - c. inaccurate
21. Anything of, relating to, or containing numbers:
- a. **numeric**
 - b. numbering
 - c. digit

22. A punctuation mark used to enclose arguments:
- a. **parentheses**
 - b. semicolon
 - c. hashtag
23. A program that translates the source code (e.g. Java code) all at once into a language that computer can understand:
- a. **compiler**
 - b. processor
 - c. translator
24. A program that translates a source code (e.g. Python code) line-by-line into a language that computer can understand:
- a. **interpreter**
 - b. coder
 - c. translator
25. Is a piece of hardware that performs the calculations that run a computer:
- a. **processor**
 - b. calculator
 - c. operating system
26. A special format of organizing and storing data in computers:
- a. **data structure**
 - b. storage
 - c. data type
27. An independent part of a program that performs one or more tasks:
- a. **module**
 - b. software
 - c. code
28. The spaces or tabs at the beginning of a line of code:
- a. **indentation**
 - b. linear
 - c. blank

29. A data structure that contains a group of elements that are indexed according to a definite sequence with 0 being the first index:
- a. **array/ list**
 - b. index
 - c. database
30. Inserting predefined module, package, or library into your code.
- a. **import/ include**
 - b. insert
 - c. export

Appendix C: Adjusted Foreign Language Classroom Anxiety Scale¹

This survey aims to explore your opinion about using English in classes, whether it is an English language course or any course that uses English to teach in class such as your current programming course.

The following statements refer to how you feel about using English in class. Please select one answer based on your first reaction to each statement.

Items	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1. I never feel quite sure of myself when I am speaking in English					
2. I don't worry about making language mistakes when I am speaking English in class.					
3. I tremble when I know that I'm going to be called on in a class that uses only English.					
4. It frightens me when I don't understand what the teacher is saying in English.					
5. It wouldn't bother me at all to take more classes that use only English.					
6. During class that uses only English, I find myself thinking about things that have nothing to do with the course.					
7. I keep thinking that the other students are better at English than I am.					
8. I am usually at ease during tests in classes that use only English.					

Continued on next page

¹The original scale developed by Horwitz et al. [142]

9. I start to panic when I have to speak English without preparation in class.					
10. I worry about the consequences of failing my class because my English is bad.					
11. I don't understand why some people get so upset about using English in other subjects.					
12. In classes that use English, I can get so nervous I forget things I know.					
13. It embarrasses me to volunteer answers in classes that use English.					
14. I would not be nervous speaking English with native speakers					
15. I get upset when I don't understand what the teacher is correcting in English.					
16. Even if I am well prepared for using English in class, I feel anxious about it.					
17. I often feel like not going to my class that uses only English.					
18. I feel confident when I speak English in class.					
19. I am afraid that my teacher is ready to correct every language mistake I make when I speak English in class.					
20. I can feel my heart pounding when I am going to be called on in class where I have to use English.					
21. The more I study for a test of a subject that uses English, the more confused I get.					
22. I don't feel pressure to prepare very well for using English in class.					
23. I always feel that the other students speak English better than I do.					

Continued on next page

24. I feel very self-conscious about speaking English in front of other students					
25. The classes that use English move so quickly I worry about getting left behind					
26. I feel more tense and nervous in the class that uses English than in my other classes that use my native language.					
27. I get nervous and confused when I am speaking English in class.					
28. When I am on my way to classes that use English, I feel very sure and relaxed.					
29. I get nervous when I don't understand every English word the teacher says.					
30. I feel overwhelmed by the number of rules you have to learn to speak English.					
31. I am afraid that the other students will laugh at me when I speak English.					
32. I would probably feel comfortable around native speakers of English.					
33. I get nervous when the teacher asks questions which I haven't prepared to answer in English.					

- Adjusted FLCAS in Simplified Chinese is available [here](#).
- Adjusted FLCAS in Arabic is available [here](#).

Appendix D: Literature Review Analysis for Section 2.9

Au- thor(s) (Year)	Programming University Level	Programming High/ Middle/ Elementary School	Course Type and Level		Graduates or Experts in CS or related field	Scope Reviewing or Theoret- ical Study	New Programming Language or Learning Tool
			Adult Informal / Self Education Programming course	University Level CS major /related subjects and majors			
Byrne and Lyons [70] (2001)	✓						
Prasad et al. [162] (2004)	✓						
Blewett and Achmad [71] (2005)	✓						
Pillay and Jugoo [29] (2005)	✓						
Begel and Gra- ham [159] (2005)					✓		
Ivins and Ong [23](2005)				✓			
Li and Prasad [161] (2005)				✓			
Rauchas et al. [55] (2006)				✓			
Nash [59] (2006)				✓			
Carter et al. [40] (2007)	✓						
Sheard et al. [157] (2008)	✓						
Keen and Etzkorn [32] (2009)	✓						
Ford and Venema [156] (2010)	✓						
Mohsin et al. [182] (2010)	✓						

Lau and Yuen [53] (2011)		✓				✓
Lau and Yuen [168] (2011)		✓				
Anwar et al. [65] (2012)	✓					
Mauai and Temese [66] (2012)				✓		
Elazhary [110] (2012)						✓
Mhashi and Ala- keel [30] (2013)	✓					
Zusana E. Pudy- astuti and Jozua F. Palandi [54] (2014)	✓					
Guerrouj et al. [160] (2014)				✓	✓	
Horton and Craig [22] (2015)	✓					
Pal and Iyer [11] (2015)	✓					
Gülmez and Özdener [61] (2015)		✓				✓
Qian and Lehman [60] (2016)		✓				
Mladenović et al. [105] (2016)		✓				
Barlow-Jones and van der Westhuizen [72] (2016)	✓					
Ahmed et al. [17] (2016)				✓		
Bouvier et al. [39] (2016)	✓					

Xu et al. [120] (2016)					✓
Pal [92] (2016)	✓				
Dasgupta and Hill [13] (2017)			✓		
Barlow-Jones and van der Westhuizen [73] (2017)	✓				
Raj et al. [171] (2017)				✓	
Ruby and Krs- manovic [52] (2017)					✓
Ayalew et al. [74] (2018)	✓				
Hermans et al. [104] (2018)		✓			
Soosai Raj et al. [172] (2018)				✓	
Guo [10]			1		
Ben Idris and Ammar [31] (2018)	✓				
Allen et al. [189] (2018)	✓				
Qian et al. [37] (2019)		✓			
Becker [124] (2019)					✓
Reestman and Dorn [125] (2019)			✓		
Soosai Raj et al. [173] (2019)				✓	
Kourichi and Benyelles [99] (2019)					✓

Mutanu and Machoka [67] (2019)					✓				
Johnson et al. [158] (2019)	✓							✓	
Nnass [128] (2019)	✓								
Zumor and Qasem [166] (2019)					✓				
Silva et al. [114] (2020)	✓								
Mason and Seton [41] (2020)					✓				
Piech et al. [190] (2020)			✓						
Feijóo-García et al. [163] (2020)				✓					
Piech and Abu-El-Haija [200] (2020)									✓
Da Silva and Silveira [208] (2020)	✓								
Bencheckroune [95] (2020)								✓	
Simanjutak [209] (2020)					✓				✓
Obaido et al. [113] (2020)				✓					✓
Denny et al. [127] (2021)	✓								
Perera and Ahangama [201] (2021)									✓
Guzman et al. [118] (2021)	✓								
Khan and Nabi [146] (2021)					✓				

Mason and Seton [21] (2021)				✓		
Sayago and Ber- gantinos [164] (2021)			✓			
Alrasheed et al. [58] (2021)				✓		
Rein et al. [103] (2021)					✓	
Liang et al. [126] (2021)				✓		✓
Hagiwara and Rodriguez [196] (2021)		✓			✓	
Alhamami [57] (2021)				✓		
Virkki [62] (2021)					✓	
Carvalho et al. [131] (2021)	✓					✓
Hagiwara and Rodriguez [196] (2021)		✓			✓	
Espinal et al. [78] (2022)		✓				
Agarwal et al. [145] (2022)				✓		
Lei and Allen [165] (2022)					✓	
Martínez- Valencia et al. [205] (2022)				✓		✓
Goswami and Pal [115] (2022)		✓				✓
Scott and Mason [94](2022)				✓		
Wibowo and Tan- diono [56] (2022)	✓					

Appendix E: Item Statistics Result

ENTRY	TOTAL	TOTAL		MODEL	INFIT	OUTFIT	PTMEASUR-AL	EXACT MATCH	
NUMBER	SCORE	COUNT	MEASURE	S.E.	MNSQ	ZSTD	MNSQ	ZSTD	CORR. EXP. OBS% EXP% ITEM
13	39	150	1.82	.20	1.29	2.81	1.64	4.21	-.15 .30 70.0 75.3 Q13
8	42	150	1.71	.19	1.19	2.08	1.31	2.41	.03 .31 67.3 73.6 Q8
26	57	150	1.20	.18	1.10	1.53	1.18	2.17	.17 .32 63.3 66.8 Q26
25	64	150	.99	.17	1.02	.42	1.04	.65	.29 .33 66.0 64.8 Q25
21	70	150	.81	.17	1.11	1.96	1.14	2.13	.18 .33 63.3 63.7 Q21
24	73	150	.72	.17	1.09	1.80	1.11	1.72	.20 .32 57.3 63.4 Q24
11	82	150	.45	.17	1.20	3.55	1.23	3.22	.06 .32 51.3 63.7 Q11
2	84	150	.39	.17	1.01	.20	.99	-.18	.31 .32 63.3 64.0 Q2
12	85	150	.36	.17	1.06	1.12	1.06	.88	.24 .32 58.7 64.2 Q12
19	85	150	.36	.17	.97	-.55	.98	-.32	.35 .32 66.7 64.2 Q19
9	88	150	.27	.17	1.07	1.31	1.06	.75	.23 .32 60.7 64.8 Q9
6	90	150	.20	.18	.85	-2.63	.83	-2.28	.51 .31 76.7 65.3 Q6
23	92	150	.14	.18	.90	-1.66	.86	-1.80	.45 .31 67.3 65.9 Q23
27	92	150	.14	.18	.96	-.58	.94	-.76	.36 .31 67.3 65.9 Q27
30	93	150	.11	.18	1.01	.12	1.00	-.02	.30 .31 66.7 66.2 Q30
28	94	150	.08	.18	.96	-.68	.94	-.73	.37 .31 68.7 66.6 Q28
7	97	150	-.02	.18	.88	-1.83	.83	-1.91	.47 .31 71.3 67.7 Q7
5	99	149	-.10	.18	1.13	1.78	1.18	1.72	.13 .30 61.1 68.9 Q5
22	100	150	-.11	.18	1.01	.23	.98	-.22	.29 .30 66.7 69.0 Q22
29	102	150	-.18	.18	.86	-1.96	.79	-2.14	.49 .30 75.3 69.9 Q29
10	110	149	-.49	.19	1.00	.06	1.02	.19	.27 .28 75.8 74.4 Q10
20	111	150	-.50	.19	.96	-.43	.96	-.24	.33 .28 76.0 74.5 Q20
4	117	150	-.74	.20	.90	-.92	.80	-1.32	.41 .26 79.3 78.1 Q4
14	118	150	-.78	.21	.90	-.88	.82	-1.11	.40 .26 80.0 78.7 Q14
17	118	150	-.78	.21	.93	-.56	.97	-.12	.34 .26 81.3 78.7 Q17
16	119	150	-.83	.21	.87	-1.13	.73	-1.76	.46 .26 79.3 79.3 Q16
15	124	150	-1.06	.22	.90	-.71	.75	-1.33	.40 .24 82.7 82.7 Q15
18	125	150	-1.11	.23	.89	-.75	.79	-1.07	.39 .24 83.3 83.3 Q18
1	132	150	-1.51	.26	.90	-.52	.75	-1.00	.37 .21 88.0 88.0 Q1
3	132	150	-1.51	.26	.93	-.33	.80	-.76	.32 .21 88.0 88.0 Q3
MEAN	94.5	149.9	.00	.19	1.00	.1	.98	.0	70.8 71.3
P.SD	24.0	.2	.83	.02	.11	1.5	.20	1.6	9.2 7.5

Appendix F: Distractor Analysis Results

ENTRY NUMBER	DATA CODE	SCORE VALUE	DATA COUNT	%	ABILITY MEAN	P.S.D	S.E. MEAN	INFT MNSQ	OUTF MNSQ	PTMA CORR.	ITEM
13	0	0	111	74	.73	.70	.07	1.2	1.3	.15	Q13
	1	1	39	26	.48*	.78	.13	1.6	1.8	-.15	
8	0	0	108	72	.65	.73	.07	1.2	1.2	-.03	Q8
	1	1	42	28	.70	.73	.11	1.3	1.4	.03	
26	0	0	93	62	.57	.59	.06	1.0	1.0	-.17	Q26
	1	1	57	38	.82	.89	.12	1.2	1.3	.17	
25	0	0	86	57	.48	.61	.07	1.0	1.0	-.29	Q25
	1	1	64	43	.91	.80	.10	1.1	1.1	.29	
21	0	0	80	53	.54	.65	.07	1.1	1.1	-.18	Q21
	1	1	70	47	.80	.78	.09	1.1	1.2	.18	
24	0	0	77	51	.52	.64	.07	1.1	1.1	-.20	Q24
	1	1	73	49	.81	.79	.09	1.1	1.2	.20	
11	0	0	68	45	.62	.67	.08	1.2	1.2	-.06	Q11
	1	1	82	55	.70	.77	.09	1.2	1.2	.06	
2	0	0	66	44	.40	.61	.08	1.0	1.0	-.31	Q2
	1	1	84	56	.87	.75	.08	1.0	1.0	.31	
12	0	0	65	43	.46	.70	.09	1.1	1.1	-.24	Q12
	1	1	85	57	.82	.72	.08	1.0	1.0	.24	
19	0	0	65	43	.37	.65	.08	1.0	1.0	-.35	Q19
	1	1	85	57	.89	.70	.08	1.0	1.0	.35	
9	0	0	62	41	.47	.60	.08	1.1	1.0	-.23	Q9
	1	1	88	59	.80	.78	.08	1.1	1.1	.23	
6	0	0	60	40	.21	.60	.08	.8	.8	-.51	Q6
	1	1	90	60	.96	.65	.07	.9	.9	.51	
23	0	0	58	39	.25	.54	.07	.9	.8	-.45	Q23
	1	1	92	61	.92	.71	.07	.9	.9	.45	
27	0	0	58	39	.33	.59	.08	.9	.9	-.36	Q27
	1	1	92	61	.87	.73	.08	1.0	1.0	.36	
30	0	0	57	38	.38	.63	.08	1.0	1.0	-.30	Q30
	1	1	93	62	.84	.73	.08	1.0	1.0	.30	
28	0	0	56	37	.32	.63	.09	1.0	.9	-.37	Q28
	1	1	94	63	.87	.70	.07	1.0	1.0	.37	
7	0	0	53	35	.20	.54	.08	.8	.8	-.47	Q7
	1	1	97	65	.92	.69	.07	.9	.9	.47	
5		***	1	1#	.02	.00				-.07	Q5
	0	0	50	34	.53	.63	.09	1.1	1.2	-.13	
	1	1	99	66	.73	.76	.08	1.1	1.1	.13	
22	0	0	50	33	.36	.60	.09	1.0	1.0	-.29	Q22
	1	1	100	67	.81	.74	.07	1.0	1.0	.29	
29	0	0	48	32	.14	.55	.08	.8	.7	-.49	Q29
	1	1	102	68	.91	.67	.07	.9	.9	.49	
10		***	1	1#	.02	.00				-.07	Q10
	0	0	39	26	.34	.70	.11	1.0	1.0	-.27	
	1	1	110	74	.78	.71	.07	1.0	1.0	.27	
20	0	0	39	26	.26	.67	.11	.9	1.0	-.33	Q20
	1	1	111	74	.81	.69	.07	1.0	1.0	.33	
4	0	0	33	22	.09	.56	.10	.8	.8	-.41	Q4
	1	1	117	78	.82	.69	.06	.9	.9	.41	
14	0	0	32	21	.10	.61	.11	.8	.8	-.40	Q14
	1	1	118	79	.82	.68	.06	.9	.9	.40	
17	0	0	32	21	.19	.70	.13	.9	1.0	-.34	Q17
	1	1	118	79	.79	.68	.06	.9	.9	.34	
16	0	0	31	21	.01	.49	.09	.8	.7	-.46	Q16
	1	1	119	79	.83	.68	.06	.9	.9	.46	
15	0	0	26	17	.03	.54	.11	.8	.7	-.40	Q15
	1	1	124	83	.80	.69	.06	.9	.9	.40	
18	0	0	25	17	.03	.62	.13	.8	.8	-.39	Q18
	1	1	125	83	.79	.68	.06	.9	.9	.39	
1	0	0	18	12	-.07	.59	.14	.7	.7	-.37	Q1
	1	1	132	88	.76	.69	.06	.9	.9	.37	
3	0	0	18	12	.02	.59	.14	.8	.8	-.32	Q3
	1	1	132	88	.75	.70	.06	1.0	1.0	.32	

Appendix G: Published Paper:



The Influence of Foreign Language Classroom Anxiety on Academic Performance in English-based CS1 Courses

Suad Alaofi*
salaofi@su.edu.sa
University College Dublin
Dublin, Ireland

Seán Russell
Sean.Russell@ucd.ie
University College Dublin
Dublin, Ireland

ABSTRACT

The use of the English language as a medium of instruction is becoming the norm in many higher education institutions around the world. This means all subjects, especially in science disciplines, are taught in English regardless of the native language of the students. For decades research has highlighted the negative impact that foreign language classroom anxiety can have on students' academic performance in foreign language classes. However, research on the effect of this type of anxiety on the students' academic performance in other classes is still limited. This research investigates the impact of foreign language classroom anxiety on the academic performance of non-native English-speaking students' in CS1 courses. A total of 160 students from two universities in different countries participated in this study. The main aim is to examine the correlation between the foreign language classroom anxiety level and the students' academic performance in assignments and final exams as well as the overall score in the CS1 courses. The results show a significant low negative impact of foreign language classroom anxiety on their academic performance in programming assignments. The research also discusses the impact of foreign language classroom anxiety based on gender and nationality. Moreover, the correlations between the students' academic performance and six different factors related to foreign language classroom anxiety have been investigated.

CCS CONCEPTS

• Social and professional topics → CS1; Computer science education.

KEYWORDS

CS1, programming, non-native English-speaking students, predicting, foreign language classroom anxiety

ACM Reference Format:

Suad Alaofi and Seán Russell. 2022. The Influence of Foreign Language Classroom Anxiety on Academic Performance in English-based CS1 Courses. In *The United Kingdom and Ireland Computing Education Research (UKICER) Conference (UKICER2022)*, September 1–2, 2022, Dublin, Ireland. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3555009.3555020>

*Also with Shaqra University.



This work is licensed under a Creative Commons Attribution International 4.0 License.

UKICER2022, September 1–2, 2022, Dublin, Ireland
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9742-1/22/09.
<https://doi.org/10.1145/3555009.3555020>

1 INTRODUCTION

English is the official language of international business, tourism, technology, etc. There are different reasons for this, including globalisation and the need for a shared medium of communication [41]. Universities around the world seek to internationalise by adopting English as a Medium of Instruction (EMI) [21]. In many cases, students have to study the English language simultaneously with studying other subjects through English [5]. Many of those students face issues relating to language skills including reading, writing, listening and speaking that can affect their academic performance in different disciplines [25]. There are many benefits of adopting EMI including the international recognition of the institution and improving students' career opportunities [21]. However, researchers highlighted many disadvantages of such a system, one of them being the psychological pressure students experience in class or during the course [40]. This pressure can be a result of speaking anxiety, fear of negative evaluation or lack of confidence when using a foreign language in class [43], all of which are considered factors of foreign language anxiety [45].

The term foreign language anxiety (FLA) refers to the stress and worry caused by learning or using a foreign language and it is well known in the area of foreign language education [27]. In foreign language classrooms, the learners can develop foreign language anxiety for different reasons including interpersonal factors, learner or teacher beliefs about language learning, the interaction between the learner and the teacher as well as the classroom setting [47]. This type of foreign language anxiety is usually referred to in the literature as Foreign Language Classroom Anxiety (FLCA). For decades, many researchers acknowledged the impact of FLCA on the students' academic performance in foreign language classes. Existing research has shown that regardless of country, FLCA is negatively correlated with students' language achievements in language classes [44]. The same sources of FLCA in the foreign language classroom also exist in EMI classes regardless of the subject of the courses [43]. This fact encouraged a number of researchers to investigate the level of FLCA among students in EMI. They all concluded that students in EMI classes suffer to some extent from FLCA [8, 10, 43]. However, the research on its impact on the students' academic performance in such classes is still limited.

This research aims to investigate the correlation between FLCA and Non-native English-speaking (NNES) students' academic performance in EMI courses, particularly, CS1 courses. The questions that this research seeks to answer are:

- RQ1: To what extent does FLCA correlate with the academic performance of NNES students in EMI CS1 courses?
RQ2: Does FLCA have the same impact on the academic performance of NNES students of different gender?

RQ3: Does FLCA have the same impact on the academic performance of NNES students from different countries?

RQ4: Do any components of FLCAS have a higher impact on the academic performance of NNES students in CS1?

2 LITERATURE REVIEW

University-level students have many sources that can create a high level of anxiety, especially first-year students [13]. A new study environment, new social life, and new subjects to study are only a few examples of those sources. For a computer science major, programming is one of the main and most challenging courses. Many of those challenges facing novice programmers are well known in the CS Education Community and being an English learner adds another layer of difficulty [38]. This research focuses on FLCA and its impact on first-year NNES students' academic performance in the CS1 courses.

Given the lack of studies on the FLCA within the field of computer science, this section provides an overview of related work on different types of anxiety impacting computer science students. Moreover, this section presents some of the research investigating the impact of FLCA on students' language achievement. In a more related context, the work of researchers who explored the FLCA topic in EMI programs is also presented in this section.

2.1 Programming, Computer, Math, and Exam Anxiety

For CS students there are many causes for anxiety, including programming, computer usage, mathematics, and exams [32]. For example, Chang's study [7] found that increasing programming task difficulty results in a significant increase in students' anxiety levels during a programming class. Scott et al. [39] state that students experience heightened anxiety when facing a programming error in their code. Computer programming performance has been found to be negatively related to mathematical and programming anxiety [33].

While many assume that CS students are very comfortable dealing with computers, Doyle et al. [12] found that even final-year CS students feel anxious when using a computer to complete some tasks. Nolan and Bergin [32] concluded that test anxiety has a negative impact on the learning ability of programming students.

In summary, CS students, especially programming students, could experience a high to moderate level of anxiety associated with mathematics, tests, and programming itself, all of which could negatively impact their academic performance.

2.2 Foreign Language Anxiety

Foreign language anxiety as defined by MacIntyre and Gardner [27], is 'the feeling of tension and apprehension specifically associated with second language contexts, including speaking, listening, and learning'. Horwitz et al. [17] state that Foreign language anxiety is a type of anxiety mainly associated with the use of a second language and can be measured. It is generally believed that anxiety affects the students' abilities to learn new information, processing it, as well as express the learned knowledge [26]. Na [31] emphasises the impact of FLA on the students' learning achievements, stating that "the learners with high anxiety often get low achievement

and low achievement makes them more anxious about learning". The meta-analysis conducted by Botes et al. [4] which included the results found by 67 studies confirms the negative impact of FLA on all categories of academic achievement of the students in foreign language classes.

However, the impact of FLA on the students' academic performance has mostly been investigated in the context of foreign language classrooms. Very few studies address Foreign language anxiety in a more general context, such as higher education programs that use EMI in non-English speaking countries. Suzuki [43] measures students' anxiety in speaking English at Japanese universities that use EMI. The result of this study shows that students tend to be more anxious about using English at the start of the semester, but their anxiety level reduces by its end. However, students who report a high level of FLA stay highly anxious throughout the semester. This result is quite consistent with Cheng and Erben's [8] study, which has found that Chinese students' anxiety levels related to the English language decrease when they stay in the U.S. longer. These two studies provide evidence of the existence of FLA in a context wider than foreign language classrooms. However, it does not address the impact of this type of anxiety on students' academic achievements. A more inclusive study at the business school in South Korean University shows the negative influence of English language anxiety on students' achievements in EMI courses [10]. It also indicates that FLA is related to class avoidance and greater difficulties in understanding the course's learning materials.

Most of those previous studies adopted the foreign language classroom anxiety scale (FLCAS) designed by Horwitz et al. [17] to measure the level of FLA of the students in class. According to Horwitz et al. [17], this scale covers three types of anxieties related to FLA including; communication apprehension, fear of negative evaluation, and test anxiety. Even though it is widely used and considered as a reliable tool to measure FLCA, the scale developers failed to provide a clear and specific explanation of the underlying components of the scale [1, 45]. In other words, the FLCAS measures the general level of FLA students' experience in the class, but does not explain which items in the scale can be used to measure each factor or component. This encouraged many researchers to conduct exploratory factor analysis studies in order to understand the underlying components of the FLCAS [1, 9, 24, 28, 29, 34, 45, 46]. As a result, a number of different factors have been identified such as speech anxiety, fear of negative evaluation, fear of failing the class, comfortableness in speaking with native speakers, low self-confidence in speaking English, fear of not understanding and communication apprehension. Each of those factors can be measured using some of the original FLCAS items.

Many studies explored the correlation between FLCAS and static factors such as gender, age, native language, and nationality [29]. Among those static factors, gender was the most explored variable, however, conflicting results have been reported. In most cases, female students reported a higher level of FLCA compared to their male peers [3, 11, 35], but in a smaller number of studies, male students reported higher levels of FLCA [6, 15, 30]. However, despite the differences in the FLCA level reported by males and females in other studies, no significant correlations have been detected between gender and FLCA [1, 29].

The majority of previous studies focused on FLA in foreign language classrooms from different perspectives. However, the same source of anxiety exists in any other class that uses EMI [43]. The reason behind this is that students in these classes use a second language to learn different subjects, communicate with their teachers and classmates, and complete the course assignments and exams. Nevertheless, there is a lack of research on the impact of FLA on teaching a specific subject, such as programming, to students through a foreign language. Hence, this research aims to fill this gap in the literature by investigating possible correlations between FLA and the academic performance of NNES students.

3 RESEARCH METHODS

3.1 Foreign Language Classroom Anxiety Scale

This study adopted the scale developed by Horwitz et al. [17] to measure the level of FLCA. The FLCAS is a 33-item, five-point Likert scale in which the students were asked to choose one of the following options: strongly agree, agree, neither agree nor disagree, disagree, or strongly disagree to answer each item. For each item, a score was given ranging from 1 for strongly disagree to 5 for strongly agree. To analyse the results correctly, the negatively worded items were assigned a reversed order of scoring so that a higher score would indicate a higher anxiety level, where the maximum score is 165 [1].

Since the original FLCAS focuses on foreign language classes, it has been slightly adjusted to test the students' level of anxiety when using English in a programming class. For example, the original phrasing of Item 2 in the FLCAS is "I don't worry about making mistakes in English class"; the rephrased Item 2 that has been used in this research is "I don't worry about making language mistakes when I am speaking English in class".¹ It also has been translated and reviewed by native speakers of each language. The survey data has been collected through an online platform in the middle of the semester. In this study, Cronbach's Alpha was calculated at $\alpha = 0.96$, which indicates that the adjustments that have been done to the scale did not impact its reliability.

Many researchers have explored the components of the FLCAS in order to understand the different factors and components related to the FLA [1, 9, 24, 28, 29, 45, 46]. In this research, instead of conducting an exploratory factor analysis, six factors have been identified based on previous research in the area (Table 1). Those factors include (1) uncomfortableness speaking with native speakers, (2) lack of confidence in speaking English in class, (3) speech anxiety, (4) fear of negative evaluation, (5) fear of failing the class as a consequence of not being good at English, and (6) anxiety about not understanding everything taught in English.

3.2 Participants

The participants were two groups of first-year students attending CS1 classes in computer and engineering schools at two universities in China and Saudi Arabia. All participants are English learners who are attending English language classes simultaneously with their CS1 courses. In total, 167 students participated in the FLCAS questionnaire, however, 7 students' FLCAS results were excluded from

¹The complete adjusted version of FLCAS is available here.

Factor	FLCAS Items
Uncomfortableness speaking with native speakers	11, 14, 32 [1, 28]
Lack of confidence in speaking English in class	1, 7, 10, 18, 23, 28 [9, 29, 46]
Speech anxiety	1, 3, 4, 8, 12, 16, 21, 24, 27, 29, 31, 33 [1, 28]
Fear of negative evaluation	3, 13, 31 [1, 24, 28]
Fear of failing the class	10, 15, 22, 25 [1, 28, 45]
Anxiety about not understanding everything taught in English	4, 5, 6, 15, 17, 22, 29 [46]

Table 1: FLCAS Factors

the correlation analysis because of missing academic performance data².

In both institutions in this research, CS1 courses are designed to be provided in English including lectures, learning materials, assessments and exams. One significant difference between the two groups is the instructors, the instructors of the first group, the Chinese students, are native English speakers while the instructors of the second group, Saudi students, share the same native language with the students. This leads to a divergence as instructors in the Saudi group are capable and willing to respond to questions from the students in their native language, while instructors in the Chinese group are unable to do the same.

Native Chinese speakers represent approximately 70% of the participants and native Arabic speakers 30%. Both groups are between 18-23 years old, 110 identified themselves as male and 47 as female while 3 preferred not to identify their gender. 132 of the 160 students also took part in a standard English language test [14]. The maximum score in this test is 100 and it can distinguish between six levels of English proficiency starting from beginner to proficient. In this test, the Chinese students' average score shows that they have an upper-intermediate proficiency level ($\mu = 55.2$). Whereas Saudi students' average score shows that they have an intermediate proficiency level ($\mu = 42.5$).

4 RESULTS

4.1 Overall FLCA Correlation Results

The Pearson correlation analysis results for all participants' data show a significant negative low correlation between overall FLCAS, which represents the general level of FLA students' experience in class, and the students' academic performance in programming assignments ($r = -.189, P = .017$). However, no correlation was found between the overall FLCAS and the academic performance in final exams and CS1 class overall score.

At the group level, the correlation analysis results for both groups show no significant correlation between the overall FLCA and their academic performance in programming assignments, final exams and CS1 class overall score. Nevertheless, Chinese students' academic performance in assignments shows a higher correlation with FLCAS ($r = -.184, p = .052$) than Saudi students.

²The researchers received ethical exemptions for the study conducted in this paper with reference:LS-E-20-129-Alaofi-Russell from University College Dublin and Ethics Appl.202001 from Shaqra University - College of Computing and Information Technology

	Assignments	Final Exam	CS1 Class
All students ($n = 160$)	-.189*	-.025	.008
China ($n = 113$)	-.184	-.028	-.021
Saudi Arabia ($n = 47$)	-.101	-.078	-.066
Female ($n = 47$)	-.056	-.102	.121
Male ($n = 110$)	-.200*	.021	-.074

*Statistically significant ($p < 0.05$)

Table 2: Overall FLCAS correlation results

There was no significant difference observed when the results were compared based on gender $t(160) = 1.72, p = 0.08$. This is despite the fact that students identifying as female show a slightly higher level of FLCA ($M = 100.7, SD = 26.6$) than students who identified as male ($M = 93.4, SD = 23.7$). While FLCA appears to be reported at a lower level for male students, the correlation between this group and their academic performance in assignments was the highest significant correlation observed in this study ($r = -.200, p = .037$).

Looking closer at the average FLCA levels by group and gender, shows that male participants responses are generally the same regardless of which group they are from. However, as shown in Figure 1, for female participants this is not the case. Chinese female participants report a higher level of FLCA than their male counterparts. This result is consistent with many previous studies that suggested that female students experience a higher level of FLCA in foreign language classes than their male peers [3, 11]. Conversely, for the Saudi group, this situation is reversed. Saudi female participants report a lower level of FLCA than their male counterparts. While this result is consistent with some other previous work [6, 15, 30], it is generally not the case.

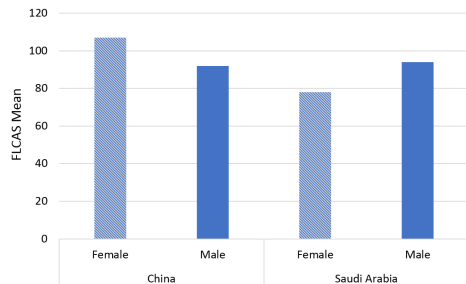


Figure 1: FLCAS Mean by Country by Gender

4.2 FLCAS Factors Analysis Results

Uncomfortableness Speaking with Native Speakers (USNS): As the name suggests, this factor measures the level of FLCA the students experience when they communicate with a native speaker. The Pearson correlation analysis results (Table 3) show a low significant negative correlation between students' academic performance in programming assignments and the level of uncomfortableness speaking with native speakers ($r = -.164, p = .039$). When looking

at the same factor correlation results with programming assignments for each group by nationality, Chinese students present a significant negative correlation ($r = -.199, p = .03$) while Saudi students show no correlation with any of the academic performance aspects.

Lack of Confidence in Speaking English in Class (LCSEC): This factor is based on self-confidence, self-esteem and nervousness in classes taught in English. Again, the results in Table 3 show a low but significant negative correlation between lack of confidence in speaking English in class and students' academic performance in CS1 assignments ($r = -.180, p = .023$). At the level of group, Chinese students show a low but significant correlation between LCSEC and academic performance in programming assignments ($r = -.187, p = .047$). The correlation between the same variables in the case of Saudi students is low and not significant.

Speech Anxiety (SA): Most of language learners experience some form of speech anxiety, which has been described as a 'mental block' during oral communication [19]. In this study, the correlation analysis results of all participants show a significant negative impact of speech anxiety on academic performance in programming assignments ($r = -.197, p = .013$). However, no significant correlations have been detected between the same factor and academic performance on the level of groups (Table 3).

Fear of Negative Evaluation (FNE): Fear of negative evaluation as described by Horwitz et al. [17] is "apprehension about others' evaluations, avoidance of evaluative situations, and the expectation that others would evaluate oneself negatively". The analysis results show no statistically significant correlations between this factor and students' academic performance in CS1 in general. At the group level, it can be said that Chinese students' academic performance in assignments might have been more negatively impacted by their fear of negative evaluation ($r = -.172, p = .068$) than Saudi students ($r = .009, p = .095$). There were no correlations between this factor and both groups' academic performance in final exams and overall CS1 scores. The full correlation results are shown in Table 3.

Fear of Failing the Class as a Consequence of not being good at English (FFC): This factor is used to evaluate the level of FLCA provoked by the fear of failing the class as a consequence of not being good at English. The selected FLCAS items for this factor are closely related to the worries of getting left behind and failing class because of not being able to follow their teachers when they speak English. As shown in Table 3, a low but significant negative correlation was observed between participants' academic performance in assignments and FFC ($r = -.169, p = .032$). However, no correlations have been found between this factor and any of the academic performance aspects on the level of groups.

Anxiety about not Understanding Everything Taught in English (AUE): Previous studies stated that understanding and speaking a specific foreign language are the most aspects that can provoke FLCA in students attending a class that uses this particular language [24]. Again, there is a low significant negative correlation between this factor and all participants' academic performance in CS1 assignments but not on the level of groups (Table 3).

Finally, no significant correlation was found between FLCA levels and students' English proficiency scores in general or in the level of groups (i.e., Chinese and Saudi, Female and male).

		USNS	LCSEC	SA	FNE	FFC	AUE
Assignment	All participants ($n=160$)	-.164*	-.180*	-.197*	-.151	-.169*	-.173*
	China ($n=113$)	-.199*	-.187*	-.178	-.172	-.151	-.137
	Saudi Arabia ($n=47$)	-.185	.002	-.129	.009	-.059	-.206
Final Exam	All participants ($n=160$)	-.070	.017	-.005	-.017	-.017	-.069
	China ($n=113$)	-.026	-.008	-.019	-.011	-.004	-.035
	Saudi Arabia ($n=47$)	-.144	.002	-.036	-.081	-.174	-.226
CS1 Class	All participants ($n=160$)	-.104	.064	.031	.015	.039	-.056
	China ($n=113$)	-.010	-.016	-.008	-.018	.009	-.028
	Saudi Arabia ($n=47$)	-.147	.004	-.057	-.011	-.109	-.224

*Statistically significant ($p < 0.05$)

Table 3: FLCAS factors correlation results

5 LIMITATIONS AND FUTURE WORK

During the design of this research, the selected courses were traditional (i.e. face-to-face) CS1 courses. However, all selected CS1 courses were converted to online courses as a consequence of COVID-19 restrictions. Consequently, the influence of the online teaching method on the results is not clear and focused research is needed in this regard. Such influence has been noticed in the absence and expected correlation between the “Anxiety about not understanding everything taught in English” and students’ academic performance on the level of groups. The possible explanation is that the availability of the recorded lectures students can re-watch has helped them to overcome this fear. This explanation is based on a recent study on FLA in a virtual language classroom where the participants said that providing them with the recorded lectures makes them feel assured that they will not miss anything [18].

As an aim for the results to be more generalised, students from two different cultural backgrounds and native languages have been invited to participate in this research. Also, they are attending different CS1 courses with different instructors who are also from different backgrounds and native languages. However, the exact impact of those differences on the results cannot be determined. Another challenge emerged from the aiming for generalisable results, which is the fact that it is impossible to find the exact same CS1 courses being offered in different countries and institutions. The differences can be found in the structure, content, assessments, and grading system of those courses. However, the selected courses are very similar in content and topics they cover. In terms of the grading system, both institutions apply a very similar scale system that includes fail, acceptable, good, very good and excellent assessments, but with slightly different marking schemes. In this study, the grades of assignments, final exams, and overall scores for both groups have been normalised to make them more comparable. Still, there is a possible impact of these differences on the results that cannot be easily distinguished.

The researchers investigated six FLCAS factors that appeared in different previous exploratory studies. However, Horwitz [16] suggested that FLCAS factors could be different from one group to another for many different reasons including cultural and native language backgrounds. So, the FLCAS factors presented in this study may not cover all factors that could impact NNES students in EMI CS1 courses. Future studies may consider conducting an exploratory factor analysis prior to investigating correlation analysis

with students’ academic performance. Moreover, based on previous research English language proficiency could have an influence on the level of FLCA. In this research, this factor has been highlighted but not controlled. Thus, the impact of the student’s proficiency in English on FLCA levels in this research is not clear and further investigation is needed. Finally, it is impossible to control the self-directed learning resources students could access in their native language, these resources would have an influence on their academic performance. This is added to other hidden variables that could have an impact on the students’ outcomes that cannot be controlled such as life circumstances, especially during the COVID-19 pandemic.

6 DISCUSSION

One of the main aims of this research is to investigate the correlation between FLCA and the academic performance of NNES students in EMI CS1 classes (RQ1). A significant low negative correlation was observed between FLCA and students’ academic performance in programming assignments. While no significant correlation was detected between FLCA and final exam and overall CS1 score. This negative impact of FLCA on students’ achievements is generally consistent with many previous studies, particularly in the field of language education [4]. However, these results contrast with the work of Alghazo and Al-Wadi [2] where they found no correlation between FLCA and students’ Mathematics achievements.

As the final grade of a CS1 course is determined by a number of different components that are completed and evaluated in different contexts, FLCA levels were compared against these results too. It was expected that assessment in the form of a terminal exam completed in English on paper would be more negatively impacted than assignments where the final deliverable would be primarily code. Analysis instead displayed a higher significant negative correlation between FLCA and assignment results. There is no definite explanation for this result but it could be the differences in the nature of exams and assignments. Exams usually contain a number of separate questions, so failing to understand one question will not impact the rest. Unlike exams, an assignment is a single programming problem that usually contained a series of connected components that need to be fully understood in order to produce the final code. In this research, It is possible that a number of students faced difficulties in understanding the assignments but they were too anxious to ask questions in English which negatively impacted

their grades. This is highly consistent with the result of Khan and Nabi's research about the use of English versus native language in programming classes [20]. They concluded that even when the students show a high level of English writing and reading skills they prefer oral communication to be in their native language and hesitate to engage if they were forced to speak in English.

The next aim was to determine if there was any difference in the observed impact based on gender (RQ2). Again, FLCA levels were correlated with results in the classes overall and the individual components of the classes. In this case, significant differences can be observed between the male and female participants. A statistically significant moderate correlation is shown between the FLCA levels of male participants and their results in assignments.

This finding is highly consistent with Koul et al. and Park and French [22, 35] who found that despite females reporting a higher FLA they performed significantly better in learning English than males. In both studies, researchers attributed this finding to the higher motivation female students have to learn foreign languages as well as increased parental and social support they may receive to help them overcome their anxiety.

To address the third aim of the study (RQ3), the data was analysed based on the participant groups. In this case, no statistically significant correlations were observed between overall FLCAS levels and students' academic performance. There is a weak correlation between FLCA levels and assignment results displayed by the Chinese students. Interestingly, when compared based on gender, male participants displayed very similar levels of FLCA while female participants tended to be higher in the Chinese cohort and lower in the Saudi cohort.

The final aim of the study relates to the factors of the FLCAS that may have a greater impact on students (RQ4). The analysis carried out was also performed for all participants as well as on the level of the country of participants. The researchers decided to perform the analysis based on the group level as a further investigation of the impact of FLCA on students from different nationalities (RQ3). In general, five out of six factors were found to be negatively correlated with all participants' academic performance in programming assignments. Correlations were identified with the factors "*Uncomfortableness speaking with native speakers*", "*Lack of confidence in Speaking English in Class*", "*Speech Anxiety*", "*Fear of Failing the Class as a Consequence of not being good at English*", and "*Anxiety about not Understanding Everything Taught in English*". No such correlations have been detected between any of those factors and final exams or overall CS1 score.

On the level of groups, none of the factors had a statistically significant correlation with the results of students from Saudi Arabia while the performance of students from China in programming assignments has low to moderate significant correlations with two out of six studied factors. Correlations were identified with the factors "*Uncomfortableness speaking with native speakers*", and "*Lack of confidence in Speaking English in Class*". The difference between the two groups of participants with respect to performance in assignments is not immediately apparent and as such conclusions are speculative in nature and require further investigation.

The primary explanation for this observation is based on the theory that FLCA and more specifically the above factors have negatively impacted the help-seeking behaviours of the participants.

This would disproportionately affect the participants from China as they are unable to ask questions or otherwise communicate with their instructors in their native language, while participants from Saudi Arabia share the same native language as their instructors.

In two related studies on teaching operating systems using a mix of English and students' native languages, the researchers find a beneficial effect on students' engagement even if it does not impact performance [37, 42]. Moreover, other studies have observed that when EMI is used in programming classes, NNES students often score lower, unlike when the medium of instruction is in their native language [23, 36]. Consequently, in this study, it is possible that the ability of students to use their native language in class has helped Saudi students engage more in class and seek help and explanation for assignments from their instructors which results in less impact of FLCA on their academic performance.

7 CONCLUSION

In conclusion, this research investigated the negative impact of foreign language classroom anxiety on NNES students attending EMI CS1 courses. The overall FLCAS levels show a low but significant negative correlation with programming assignments but not final exams and overall CS1 scores. Moreover, five out of six studied components of FLCAS show a significant correlation with the students' academic performance in programming assignments. However, on the group level, only two components show significant correlations with academic performance in programming assignments for one of the studied groups. For this reason, the negative influence of FLCA on NNES students' academic performance in EMI CS1 courses cannot be completely excluded. However, more investigation should be considered to confirm these results.

In terms of the implications, the finding of this study can be helpful for EMI CS1 course instructors as well as institutions that offer those courses for NNES. For example, instructors are advised to consider that providing recorded lectures can be beneficial for their students to overcome the stress of missing important information in class. Also, if instructors are aware of the possible impact of FLCA on students' performance, particularly in assignments, they could use different learning strategies to help them seek help and ask more questions. At the institutional level, the availability of instructors or teaching assistants who share the same native language with the students may reduce the negative impact of FLCA on NNES students learning experience. Yet, this is not always possible, especially in classes where there are students from mixed backgrounds and native languages. A more helpful approach to this issue may include raising the awareness of the negative impact of FLCA on learning among faculty members and students. In brief, institutions should consider new strategies that ensure a better learning environment for all students including NNES students. Strategies and techniques used in the field of foreign language teaching in may prove effective in reducing the negative effects of FLCA.

REFERENCES

- [1] Yukie Aida. 1994. Examination of Horwitz, Horwitz, and Cope's construct of foreign language anxiety: The case of students of Japanese. *The modern language journal* 78, 2 (1994), 155–168.
- [2] Yazan Alghazo and Hasan Al-Wadi. 2019. English Language Anxiety and Its Effect on Students' Mathematics Achievement. *International Education Studies* 12, 12 (2019), 130–140.

- [3] Elias Bensalem. 2018. Multilingualism and foreign language anxiety: the case of Saudi EFL learners. *Learning and Teaching in Higher Education: Gulf Perspectives* 15, 2 (2018), 47–60.
- [4] Eloise Botes, Jean-Marc Dewaele, and Samuel Greiff. 2020. The foreign language classroom anxiety scale and academic achievement: An overview of the prevailing literature and a meta-analysis. *Journal for the Psychology of Language Learning* 2, 1 (2020), 26–56.
- [5] Dennis Bouvier, Ellie Lovellette, John Matta, Bedour Alshaigi, Brett A. Becker, Michelle Craig, Jana Jackova, Robert McCartney, Kate Sanders, and Mark Zarb. 2016. Novice Programmers and the Problem Description Effect. In *Proceedings of the 2016 ITICSE Working Group Reports*. Association for Computing Machinery, New York, NY, USA, 103–118. <https://doi.org/10.1145/3024906.3024912>
- [6] Seyit Ahmet Capan and Harun Simsek. 2012. General foreign language anxiety among EFL learners: A survey study. *Frontiers of language and teaching* 3, 11 (2012), 116–124.
- [7] Sheeson E Chang. 2005. Computer anxiety and perception of task complexity in learning programming-related skills. *Computers in Human Behavior* 21, 5 (2005), 713–728.
- [8] Rui Cheng and Antony Erben. 2012. Language anxiety: Experiences of Chinese graduate students at US higher institutions. *Journal of Studies in International Education* 16, 5 (2012), 477–497.
- [9] Yuh-show Cheng, Elaine K Horwitz, and Diane L Schallert. 1999. Language anxiety: Differentiating writing and speaking components. *Language learning* 49, 3 (1999), 417–446.
- [10] Seungwoo Chun, Hyondong Kim, Chan-Kyoo Park, Karin McDonald, Oh Sun Ha, Dae Lyong Kim, and Seon Min Lee. 2017. South Korean students' responses to English-medium instruction courses. *Social Behavior and Personality: an international journal* 45, 6 (2017), 951–965.
- [11] Jean-Marc Dewaele and Livia Dewaele. 2017. The dynamic interactions in foreign language classroom anxiety and foreign language enjoyment of pupils aged 12 to 18. A pseudo-longitudinal investigation. *Journal of the European Second Language Association* 1, 1 (2017), 12–22.
- [12] E. Doyle, I. Stamouli, and M. Huggard. 2005. Computer anxiety, self-efficacy, computer experience: an investigation throughout a computer science degree. In *Proceedings Frontiers in Education 35th Annual Conference*. IEEE, IN, USA, S2H–3.
- [13] Natalie Eckberg, Aileen M Pidgeon, and Heidi Magyar. 2017. Examining the psychosocial and academic factors predicting depression and anxiety symptomatology across first year and later year university students. *European Scientific Journal* 13, 17 (2017), 1–16.
- [14] EF-Education. 2021. English levels and English proficiency scores: EF SET. <https://www.efset.org/english-score/>
- [15] Diana Chitra Hasan and Sitti Fatimah. 2014. Foreign language anxiety in relation to gender equity in foreign language learning. In *Equality in Education*. SensePublishers, Rotterdam, 183–193.
- [16] Elaine K Horwitz. 2016. Factor structure of the foreign language classroom anxiety scale: Comment on Park (2014). *Psychological reports* 119, 1 (2016), 71–76.
- [17] Elaine K Horwitz, Michael B Horwitz, and Joann Cope. 1986. Foreign language classroom anxiety. *The Modern language journal* 70, 2 (1986), 125–132.
- [18] Mohammad Tanvir Kaisar and Sabrina Yasmin Chowdhury. 2020. Foreign Language Virtual Classroom: Anxiety Creator or Healer?. *English Language Teaching* 13, 11 (2020), 130–139.
- [19] Said Muhammad Khan. 2015. Influence of Speech Anxiety on Oral Communication Skills among ESL/EFL Learners. *Advances in Language and Literary Studies* 6, 6 (2015), 49–53.
- [20] Taj Muhammad Khan and Syed Waqar Nabi. 2021. English versus Native Language for Higher Education in Computer Science: A Pilot Study. In *21st Koli Calling International Conference on Computing Education Research*, 1–5.
- [21] Jane Knight. 2013. The changing landscape of higher education internationalisation – for better or worse? *Perspectives: Policy and Practice in Higher Education* 17, 3 (2013), 84–90. <https://doi.org/10.1080/13603108.2012.753957>
- [22] Ravinder Koul, Laura Roy, Sittichai Kaewkuekool, and Suthee Ploisawaschai. 2009. Multiple goal orientations and foreign language anxiety. *System* 37, 4 (2009), 676–688.
- [23] Wilfred WF Lau and Allan HK Yuen. 2011. Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education* 57, 1 (2011), 1202–1213.
- [24] Meihua Liu and Jane Jackson. 2008. An exploration of Chinese EFL learners' unwillingness to communicate and foreign language anxiety. *The Modern Language Journal* 92, 1 (2008), 71–86.
- [25] Ernesto Macaro, Samantha Curle, Jack Pun, Jiangshan An, and Julie Dearden. 2018. A systematic review of English medium instruction in higher education. *Language Teaching* 51, 1 (2018), 36–76.
- [26] Peter MacIntyre and Tammy Gregersen. 2012. Affect: The role of language anxiety and other emotions in language learning. In *Psychology for language learning*. Springer, London, 103–118.
- [27] Peter D MacIntyre and Robert C Gardner. 1994. The subtle effects of language anxiety on cognitive processing in the second language. *Language learning* 44, 2 (1994), 283–305.
- [28] Barley Mak. 2011. An exploration of speaking-in-class anxiety with Chinese ESL learners. *System* 39, 2 (2011), 202–214.
- [29] Sae Matsuda and Peter Gobel. 2004. Anxiety and predictors of performance in the foreign language classroom. *System* 32, 1 (2004), 21–36.
- [30] Hugo Mejias, Ronald L Applbaum, Susan J Applbaum, and Robert T Trotter. 1991. Oral communication apprehension and Hispanics: An exploration of oral communication apprehension among Mexican American students in Texas. In *Language anxiety: From theory and research to classroom implications*, Elaine K. Horwitz and Dolly J. Young (Eds.). Pearson, New York, 87–97.
- [31] Zhao Na. 2007. A study of high school students' English learning anxiety. *The Asian EFL Journal* 9, 3 (2007), 22–34.
- [32] Keith Nolan and Susan Bergin. 2016. The Role of Anxiety When Learning to Program: A Systematic Review of the Literature. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16)*. Association for Computing Machinery, New York, NY, USA, 61–70. <https://doi.org/10.1145/2999541.2999557>
- [33] J Owolabi, P Olanipekun, and J Iwerima. 2014. Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming. *GSTF Journal on Computing* 3, 4 (2014), 1–6.
- [34] Gi-Pyo Park. 2014. Factor analysis of the foreign language classroom anxiety scale in Korean learners of English as a foreign language. *Psychological Reports* 115, 1 (2014), 261–275.
- [35] Gi-Pyo Park and Brian F French. 2013. Gender differences in the foreign language classroom anxiety scale. *System* 41, 2 (2013), 462–471.
- [36] Nelishia Pillay and Vikash R Juggoo. 2005. An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bulletin* 37, 4 (2005), 107–110.
- [37] Adalbert Gerald Soosai Raj, Kasama Ketsuriyong, Jignesh M Patel, and Richard Halverson. 2017. What Do Students Feel about Learning Programming Using Both English and Their Native Language?. In *2017 International Conference on Learning and Teaching in Computing and Engineering*. IEEE, IN, USA, 1–8.
- [38] Ivan Ruby and Bojana Krsmanovic. 2017. Does learning a programming language require learning English? A comparative analysis between English and programming languages. In *EdMedia+ Innovative Learning*. Association for the Advancement of Computing in Education (AACE), 420–427.
- [39] Michael James Scott and Gheorghita Ghinea. 2014. Measuring Enrichment: The Assembly and Validation of an Instrument to Assess Student Self-Beliefs in CS1. In *Proceedings of the Tenth Annual Conference on International Computing Education Research (Glasgow, Scotland, United Kingdom) (ICER '14)*. Association for Computing Machinery, New York, NY, USA, 123–130.
- [40] Kamran Akhtar Siddiqui. 2021. Challenges for first-year undergraduates with a non-EMI background in an EMI classroom: A case study. *The Education and science journal* 23, 8 (2021), 189–201.
- [41] Ute Smit. 2010. *English as a lingua franca in higher education*. De Gruyter Mouton, Berlin, Germany.
- [42] Adalbert Gerald Soosai Raj, Eda Zhang, Saswati Mukherjee, Jim Williams, Richard Halverson, and Jignesh M. Patel. 2019. Effect of Native Language on Student Learning and Classroom Interaction in an Operating Systems Course. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (Aberdeen, Scotland UK) (ITICSE '19)*. Association for Computing Machinery, New York, NY, USA, 499–505.
- [43] Noriko SUZUKI. 2017. Foreign language speaking anxiety in the English-medium instruction (EMI) program in a Japanese university. *CASELE Research Bulletin* 47 (2017), 11–21.
- [44] Yasser Teimouri, Julia Goetze, and Luke Plonsky. 2019. Second language anxiety and achievement: A meta-analysis–erratum. *Studies in Second Language Acquisition* 41, 2 (2019), 489–489.
- [45] Michiko Toyama and Yoshitaka Yamazaki. 2018. Exploring the components of the foreign language classroom anxiety scale in the context of Japanese undergraduates. *Asian-Pacific Journal of Second and Foreign Language Education* 3, 1 (2018), 1–27.
- [46] Tomoko Yashima, Noels Kimberly A, Tetsuhito Shizuka, Osamu Takeuchi, Shigeru Yamane, and Kiyomi Yoshizawa. 2009. The Interplay of Classroom Anxiety, Intrinsic Motivation, and Gender in the Japanese EFL Context. *Journal of Foreign Language Education and Research* 17, 3 (2009), 41–64.
- [47] Dolly Jesusita Young. 1991. Creating a low-anxiety classroom environment: What does language anxiety research suggest? *The modern language journal* 75, 4 (1991), 426–439.