Programming: Factors that Influence Success

Susan Bergin
Department of Computer Science,
National University of Ireland Maynooth,
Maynooth, Co. Kildare.
sbergin@cs.may.ie

Ronan Reilly
Department of Computer Science,
National University of Ireland Maynooth,
Maynooth, Co. Kildare.
ronan@cs.may.ie

ABSTRACT

This paper documents a study, carried out in the academic year 2003-2004, on fifteen factors that may influence performance on a first year object-oriented programming module. The factors included prior academic experience, prior computer experience, self-perception of programming performance and comfort level on the module and specific cognitive skills. The study found that a student's perception of their understanding of the module had the strongest correlation with programming performance, $r = 0.76, p < 0.01$. In addition, Leaving Certificate (LC) mathematics and science scores were shown to have a strong correlation with performance. A regression module, based upon a student's perception of their understanding of the module, gender, LC mathematics score and comfort level was able to account for 79% of the variance in programming performance results.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education

General Terms

Human Factors, Measurement, Performance

Keywords

CS1, Programming, Predictors

1. INTRODUCTION

Student retention on third-level (post high school or equivalent) Computer Science (CS) and Information Technology (IT) courses is a significant problem. Students find computer programming difficult and struggle to master the core concepts. A multi-national, multi-institutional study on the programming skills of first year CS students found that students struggled to achieve an average above 30% on assessments administered as part of their study [14]. Furthermore, introductory programming modules tend to have a very high student to lecturer ratio (100:1 or greater) and often lecturers do not know how well students are doing until after the first assessment. In general the first assessment does not take place until six or eight weeks after the module has commenced and given the typically high number of students, marking the assessments can take a considerable length of time. At this stage, it may be too late to intervene to prevent struggling students from failing. Even if intervention is possible, a lecturer is unlikely to know a student well enough or be able to identify individual student problems and therefore recognize the most suitable interventions to make.

The research documented in this paper is part of a longitudinal study on early identifiable factors that influence performance on an introductory programming module. If such factors can be identified then it may be possible to develop a tool to provide an early diagnosis of a student's likely performance on a programming module. Interested parties could use the tool to make more informed decisions on appropriate courses of action and to decide upon personalized interventions that foster a student's intellectual strengths.

2. RELATED RESEARCH

While a considerable amount of research has been carried out on factors that affect programming performance our interest is on factors that affect programming performance on an introductory third-level object-oriented programming module where such factors can be determined early in the academic year. These factors can be categorized as: (1) previous academic and computer experience, (2) cognitive skills, (3) personal information and (4) experience on the module. A brief review of some research studies in each of these categories is presented next.

Previous academic experience and programming experience have often been cited as predictors of programming success. Several studies have found that mathematical ability and exposure to maths courses are important predictors of performance on introductory computer science modules [2, 3, 4, 6, 13, 19]. Similarly, although less studied, performance in and experience of science subjects has also been shown to be important [2, 19]. Studies have also found prior programming experience and non-programming computer experience (for example, experience of computer applications, emailing, game playing and surfing the web) to be related to programming performance [9, 10, 6, 4, 18].

The role of cognitive factors in programming has also received research attention. Certain cognitive factors, including problem-solving, abstract reasoning, problem translation skills, logical ability and cognitive style have been identified as possible predictors [7, 12, 11, 15].

Numerous studies have been carried out on demographic and self-reported personal information. Some studies have examined specific attributes related to study, for example preference for working alone or in a group to solve a programming problem and encouragement or support from others to study computers [4, 2, 8].

In recent times researchers have examined the relationship between students' expectations of and experiences on an introductory computing module. A positive relationship between a student's mental models of programming and self-efficacy for programming and performance has recently been identified [18]. The grade a student expected to achieve in an introductory module was found to be the most important indicator of performance in another recent study[17]. A recent longitudinal study found that the most important predictor of students' performance on an introductory computer science course was comfort level, determined by the degree of anxiety a student felt about the course [4].

Although a considerable number of research studies have examined factors that influence programming performance, comparisons between the various studies and application of the results are difficult because the studies are carried out using different parameters. These parameters include:

- the type of students (novice to experienced programmers, academic students to employees),
- the content of the course (some courses are solely programming courses while others are introductory computing courses),
- the programming language being taught (older studies tend to be based on procedural languages),
- the educational setting (many of the research studies are based on the US educational system) and
- the reference criterion (for example continuous assessment, end-of-year exam, job performance).

In the rest of this paper we describe our study, which builds upon existing research, in particular the work of [2, 4], to identify early factors that influence performance of first year students, on an introductory object-oriented programming module, using the Java programming language. The course is based on the Irish educational system and the reference criterion is the overall result achieved on the module.

3. RESEARCH DESIGN

The introductory programming module at our university is composed of a one and a half hour Problem-Based Learning (PBL) workshop, a one and a half hour laboratory session and three one-hour lectures per week over two semesters. Students in Ireland do not study programming in secondary school and the majority of students taking this module have recently completed second level education.

Selection of factors for this study was restricted for a number of reasons, including availability of participants, length of completion time needed for each instrument and stage in year. With this in mind we attempted to examine the relationship between and the predictability of fifteen factors and performance on our introductory module. The factors fall into four broad categories:

1. previous academic and computer experience: as measured by performance in the Irish Leaving Certificate (LC) examinations in mathematics and science subjects and self-reported computer experience,
2. specific cognitive skills: as measured by an in-house cognitive test,
3. personal information: gender, age, work-style preference, encouragement from others and the number of hours per week working at a part-time job and
4. experience on the module: students own perception of how well they are doing and how comfortable they feel with the module material.

Performance on this module is based on continuous assessment (30% of the overall mark) and a final examination (70% of the overall mark). The measure of performance reported upon in this paper is the overall module mark. Continuous assessment and final examination marks render similar results and are reported in [1].

3.1 Participants

The study was carried out in the academic year 2003-2004. Students enrolled in the first year 'Introduction to Programming' module in our department voluntarily participated in this study. Ninety-six students completed the module in the academic year 2003-2004.

3.2 Instruments

Two instruments were used to collect data: a questionnaire and a custom-made cognitive test. The questionnaire collected data on the following items: (1) LC mathematics grade, (2) LC physics grade, (3) LC biology grade, (4) LC chemistry grade, (5) highest LC science grade, (6) comfort level on the module, (7) perceived understanding of the module material, (8) prior programming experience, (9) prior non-programming computer experience, (10) work-style preference (preference to work-alone or as part of a group), (11) encouragement from others to study computer science, (12) number of hours per week working at a (part-time) job. The cognitive test was developed in-house[1] and comprised items involving numerical and letter sequencing, arithmetic reasoning, problem translation skills and logical ability. In addition, information on gender, age and overall module results was available for all students taking the module.

In total 80 students (49 male, 31 female) completed the cognitive test and 30 (19 male, 11 female) students completed the survey. Both instruments were completed in the second semester of the module and data collection for both was paper-based.

4. RESULTS

An a priori analysis was carried out to verify no significant difference existed between the mean overall module scores of the class and the sample. Test assumptions on normality (Kolmogorov-Smirnov test) and the equality of variance (Levine test) were performed and a t-test on the overall results, ($t(124) = 0.795, p = 0.428$), found no significant differences between the mean scores of the class and

[1]Developed by Jacqueline McQuillan, Department of Computer Science, NUI Maynooth.

Table 1: Pearson correlations for previous academic results and performance

	LC Maths	LC Phys	LC Chem	LC Bio	High Sci
r	0.46**	0.59*	0.4	0.75*	0.48**
n	30	18	11	10	28
Female only					
r	0.72*	0.89**	0.88	0.93**	0.84**
n	11	7	4	7	11
Male only					
r	0.16	0.27	-0.04	0.9	0.16
n	19	11	7	3	17

** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

the sample. In the remainder of this section the findings on the relationship between each of the factors studied and programming performance is presented, followed by an analysis of the combination of factors that best predicts performance.

4.1 Previous academic and computer experience

To establish the relationship between previous academic experience in mathematics and science, the achievable grades for each subject were ranked, with the highest rank given to the highest possible grade and the lowest rank given to the lowest possible grade. Table 1. provides the Pearson correlations for each of these measures and notable relationships are identified. LC mathematics was found to have a statistically significant relationship with performance, $r = 0.46, p < 0.01$. LC physics was found to be moderately strong and significant, $r = 0.59, p < 0.05$ as was LC biology, $r = 0.75, p < 0.05$ for the final examination. Highest science result, which includes other less commonly studied science subjects, was also found to be statistically significant, $r = 0.48, p < 0.01$. No relationship was found between LC chemistry and performance. Secondary analysis, based on gender revealed that none of the measures were significant for male students and resulted in notably higher correlations for the female students, as shown in Table 1. A recent study on gender differences in LC examinations found that (1) more female students are taking higher level LC examinations than male students and (2) female students are outperforming male students on LC maths and physics examinations (no other science subjects were reviewed in the study) [5]. This may relate to our findings and further research is necessary.

The findings on the relationship between experience in mathematics and science subjects, and programming performance is in line with previous research findings. The strength of the correlations between LC physics scores and particularly LC biology scores and programming performance is interesting and would suggest that science in general has a significant influence on performance. However, the lack of correlation with LC chemistry appears contradictory and further research is required.

Table 2: Dichotomous values for personal factors

	Values
Gender	Male, Female
Age	Under 23, 23+
Work style preference	Individual, Group
Encouragement	Yes, No
Part-time employment	No, Yes

Table 3: Pearson correlations for comfort level and perceived understanding with performance

	Comfort level	Understanding
r	0.55**	0.76**
n	30	30
Female only		
r	0.62*	0.82**
n	11	11
Male only		
r	0.79**	0.84**
n	19	19

** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

Previous computer experience was measured by prior programming experience and previous non-programming computer experience. In both cases student responses were separated into those with previous experience and those without previous experience. Descriptive statistics for each group are given in Table 4. T-tests for independent samples were used to examine the differences between the overall module results of each group. Before each t-test was carried out assumptions of normality and equality of variance were confirmed. No significant differences were found between students with or without previous programming experience or between students with or without non-programming computer experience and performance module. Although previous research has found previous programming experience and non-programming computing experience to be indicators of success our results may be partially accounted for by the fact that students cannot study programming or application software at examination level in secondary schools in Ireland.

4.2 Specific cognitive skills

A correlation of $r = 0.31, p < 0.01$ was found between performance on the cognitive test and performance on the module. Although this result is weak, subsequent analysis found that a number of items in the test were highly correlated with programming performance. We anticipate that a redesign of the test could result in more significant findings in the future.

4.3 Personal information

Gender, age, work-style preference, encouragement by others and part-time employment were treated as dichotomous

Table 4: Comparison of the mean and standard deviation for overall results (as a percentage) grouped by: gender, age, work-style preference, encouragement by others, prior programming (Prog. exp.) and non-programming computer experience (Non-prog. comp. exp.)

		n	Mean (%)	S.D. (%)
Gender	Female	36	51	24
	Male	60	49	22
Age	Under 23	92	50	23
	23+	4	56	28
Work-style preference	Individual	12	50	22
	Group	18	43	23
Encouragement	No	21	44	21
	Yes	9	50	25
Part-time job	No	18	47	24
	Yes	12	45	20
Prog. exp.	None	25	46	23
	Some	5	44	18
Non-prog. comp. exp.	None	4	49	21
	Some	26	45	23

variables for analysis purposes. The possible values of each factor are given in Table 2. Students were grouped according to the responses they provided for each of the factors. Descriptive statistics for each group are given in Table 4. T-tests for independent samples were used to examine the differences between the overall module results for each of the factor values, for example the mean overall module result for male students was compared to the mean overall module result for female students. Before each t-test was carried out assumptions of normality and equality of variance were confirmed. In each instance, the t-tests revealed no significant differences between any of the factors and the overall results on the module. We intend to further examine the relationship between work-style preference and performance, as since the introduction of PBL workshops into the module mean scores have increased, at the top, middle and bottom levels of the class. We feel this is a result of the PBL workshops and students repeating the module appear to concur with us [16].

4.4 Experience on the module

Comfort level was measured as the cumulative response to questions on a student's understanding of programming concepts, difficulty designing programs without help and difficulty for completing lab assignments. Each question had a number of ranked answers and the cumulative rank was used to analyze comfort level. The Pearson correlations are given in table 3. Comfort level was found to be a statistically significant indicator of performance with $r = 0.516, p < .01$.

Understanding was measured by ranked responses to a single question 'How do you rate your level of understanding of the programming module?' A strong significant relationship between understanding and performance was found, $r = 0.76, p < 0.01$.

Given the earlier findings on gender differences between previous academic experience and programming performance, gender based analysis was carried out on comfort level and understanding. Comfort level was found to have a higher correlation with performance for male students.

Like the Cantwell Wilson and Shrock [4] study comfort level was found to be highly correlated with programming performance. The most significant finding, however, is the very strong correlation between a students' perception of their understanding of the programming module. As this study was carried out in the second semester we intend to conduct a further study to identify the point in time perception of module understanding becomes such a reliable indicator. If a similarly high correlation can be found early on in the module then it would be very powerful in diagnosing and subsequently mediating struggling students.

4.5 Regression Analysis

To investigate whether the various factors studied were predictive of performance on the module a number of regression analyzes were conducted. Each analysis was motivated by the literature review, the authors' experience working with first year students and the strength of the correlation coefficients generated in this study. Although, both LC biology and LC physics rendered high correlation coefficients for programming performance, neither variables were directly included in the regression models as the sample size for each was deemed too small (n=10, n=18 respectively). Instead the highest LC science result was included (n=28).

The first model was designed to determine the earliest indicators of programming performance. Consideration was given to gender, previous academic experience, cognitive test score, previous programming and non-programming computer performance, encouragement from others, work-style preference and hours working at a part-time job. Using a stepwise regression method a significant model emerged with $F(2, 27) = 7.113, p < 0.01$ with an adjusted R square =30%. Significant values were found for: LC maths ($\beta = 0.390, p = 0.021$) and gender ($\beta = -0.368, p = 0.028$).

The second model considered all of the predictors used in the first model but also included the results of the first class test. Class tests are typically the first test given to first year students and although they do not test a student's ability to design and code up a solution to a programming problem they do test a students' understanding of basic programming concepts. A stepwise regression method found a significant model of $F(2, 27) = 14.882, p < 0.001$ and adjusted R square =49%. Significant values were found for the class test ($\beta = 0.563, p = 0.000$) and LC maths ($\beta = 0.375, p = 0.01$).

The third model included the predictors from the second model but also considers the results of the first lab test. The lab test is similar to the final examination in that students are required to design and code up a solution to a programming problem. Although the lab test may be a better predictor of the overall result, a trade off takes place in that this information is not available until near the end of the first Semester and at this stage struggling students may have dropped out or given-up hope of succeeding. A stepwise regression method resulted in a significant model of $F(2, 25) = 26.38, p < 0.001$ and an adjusted R square = 65%. Significant factors were found for the first lab test ($\beta = 0.700, p = 0.000$) and highest science result ($\beta = 0.700, p = 0.000$).

The fourth model includes the predictors from the second model but takes into account a students' comfort level with the module and perceived understanding of how they are doing. Using a stepwise regression method a significant model emerged with $F(4, 23) = 26.03, p < 0.001$, adjusted R square = 79%. Significant values were found for: understanding ($\beta = 0.505, p = 0.000$), gender ($\beta = -0.494, p = 0.000$), comfort level ($\beta = 0.301, p = 0.022$), and LC maths ($\beta = 0.197, p = 0.047$). If the results of the first lab test is also considered 84% of the variance in performance can be accounted for with $F(4, 23) = 36.92, p < 0.001$.

The factors known at the start of the academic year result in a poor prediction of programming performance. The results of the first class test (model 2) and subsequently the first lab exam (model 3) results in an improved prediction ability. However, when a students' perception of their understanding of the module is considered a very strong prediction model occurs. As with the strength of the Pearson correlation coefficient for this variable a further study to determine the stage at which a student's self-perception becomes so accurate would be valuable.

5. CONCLUSIONS

This study examined the relationship and predictive ability between fifteen factors and performance on a programming module. Comfort level on the module, LC maths and LC science scores were shown to have a strong correlation with performance, with notable gender differences identified. A predictive combination of factors was found to be a student's perception of their understanding of the module, comfort level on the module, LC maths score and gender, accounting for 79% of the variance in programming performance.

The study found that the strongest relationship existed between a student's perception of their understanding of the module and programming performance. The need to understand the role of self-perception in the process and to investigate how early it becomes a reliable predictor warrants further research.

6. REFERENCES

[1] S. Bergin and R. Reilly. Discovering the likely performance of novice programmers on an introductory programming module. In *Technical Report NUIM-CS-TR-2004, Department of Computer Science, National University of Ireland, Maynooth*, 2004.

[2] 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*, pages 49–52, 2001.

[3] P. F. Campbell and G. P. McCabe. Predicting the success of freshmen in a computer science major. *Commun. ACM*, 27(11):1108–1113, 1984.

[4] B. Cantwell-Wilson and S. Shrock. Contributing to success in an introductory computer science course: a study of twelve factors. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, pages 184–188, 2001.

[5] J. Elwood and K. Carlisle. *Examining Gender. Gender and achievement in the Junior and Leaving Certifcate Examinations 2000/2001.* National Council for Curriculum and Assessment, 2003.

[6] G. E. Evans and M. G. Simkin. What best predicts computer proficiency? *Commun. ACM*, 32(11):1322–1327, 1989.

[7] D. C. Gibbs. The effect of a constructivist learning environment for field-dependent/independent students on achievement in introductory computer programming. In *Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, pages 207–211, 2000.

[8] A. Goold and R. Rimmer. Factors affecting performance in first-year computing. *SIGCSE Bull.*, 32(2):39–43, 2000.

[9] D. Hagan and S. Markham. Does it help to have some programming experience before beginning a computing degree program? In *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education*, pages 25–28, 2000.

[10] E. Holden and E. Weeden. The impact of prior experience in an information technology programming course sequence. In *Proceeding of the 4th conference on Information technology curriculum*, pages 41–46, 2003.

[11] T. R. Hostetler. Predicting student success in an introductory programming course. *SIGCSE Bull.*, 15(3):40–43, 1983.

[12] 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*, pages 110–117, 1980.

[13] R. R. Leeper and J. L. Silver. Predicting success in a first programming course. In *Proceedings of the thirteenth SIGCSE technical symposium on Computer science education*, pages 147–150, 1982.

[14] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz. A multi-national, multi-institutional study of assessment of programming skills of first-year cs students. *SIGCSE Bull.*, 33(4):125–180, 2001.

[15] R. Nowaczyk. Cognitive skills needed in computer programming. In *Annual Meeting of the Southeastern Psychological Association*, 1983.

[16] J. O'Kelly, A. Mooney, J. Ghent, P. Gaughran, S. Dunne, and S. Bergin. An overview of the integration of problem based learning into an existing computer science programming module. In *Pleasure By Learning, (PBL 2004)*, 2004.

[17] N. Rountree, J. Rountree, and A. Robins. Predictors of success and failure in a cs1 course. *SIGCSE Bull.*, 34(4):121–124, 2002.

[18] D. L. S. Wiedenbeck and V. Kain. Factors affecting course outcomes in introductory programming. In *Proceedings of the 16th Workshop on Psychology of Programming, PPIG'04*, pages 97–109, 2004.

[19] L. H. Werth. Predicting student performance in a beginning computer science class. In *Proceedings of the seventeenth SIGCSE technical symposium on Computer science education*, pages 138–143, 1986.