The Effect of Student Attributes on Success in Programming

Pat Byrne and Gerry Lyons,
Department of Information Technology
National University of Ireland, Galway,
Galway,
Ireland
{pat.byrne, gerard.lyons}@nuigalway.ie

Abstract

This paper examines the relationship between student results in a first year programming course and predisposition factors of gender, prior computing experience, learning style and academic performance to date. While the study does not suggest that any dominant attributes are related to success in programming, there are some interesting outcomes which will have implications for teaching and learning.

1 Introduction

There have been very few studies in recent years into academic success in computer programming [6]. Today, industry is keen to accept as many graduates as the academic institutions can produce, and there is an assumption (possibly due to the universality of computing in education and the advent of user-friendly interfaces) that any bright student can program. However, experience in the classroom would suggest that this is not true. Students who are proficient in many other subjects sometimes fail to achieve success in programming.

Evidence of influencing factors could be useful in supporting students who enter with a perceived disadvantage. We can also use such research to adapt our pedagogies to reflect the disparities present in any particular cohort of students.

2 Literature

Early studies focused on establishing relationships between academic performance and success in programming in order to define instruments to be used in student selection and career guidance. A number of these were developed, although none have gained universal acceptance in academia [3,8,17,12]. Such instruments normally take the form of an aptitude test administered to prospective students, and many of them have a mathematical focus. The link between mathematics ability and programming is widely accepted, although its empirical demonstration is questionable. Its inclusion in any related study is thus necessary.

The fact that the majority (approx. 72%) of students coming into computer science courses are male has been an area of concern to both educationalists and employers [2]. There appears to be a number of factors which deter girls from entering, and staying, in academic computing. These include the image of computer science as a male domain and the computing world embracing a culture which is not welcoming to women; the perceived lack of confidence amongst students despite their obvious abilities and successes; the lack of women lecturers and role models; the importance of prior computing experience which males and females receive outside (and inside) the school system [5,13]. Studies into the area reveal there is no simple solution as to why girls do not engage more in computing studies; the reasons are linked to the nature of socially defined gender roles. However, an examination of the performance of female students within the academy might reveal some clues which can encourage them to stay within the system, and how we can support and promote their endeavours.

In a review of studies attempting to predict computer competency done up to 1990, Evans and Simkin make a specific case for the inclusion of cognitive factors in any study of this kind [4]. They found that "... few of the demographic, academic, computer exposure or behavioural variables were particularly strong predictors of class performance...". One such cognitive aspect which is easy to measure, has particular import when attempting to establish predictors in order to improve the educational environment, and which has a recognised acceptance is that of Kolb's Learning Style Inventory (LSI) [9].

The LSI is based on the premises that learning is a four-stage cycle involving four adaptive learning modes –

ITiCSE 2001 6/01 Canterbury, UK
© 2001 ACM ISBN 1-58113-330-8/01/06...$5.00

concrete experience, reflective observation, abstract conceptualisation and active experimentation. Each individual has a preferred approach to learning, going from abstract to concrete concepts, and as such can be considered as entering the continuum in one of the quadrants shown in the learning cycle (Figure 1).

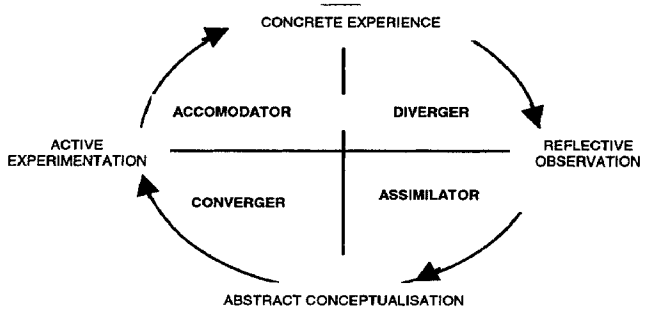

Figure 1 Learning Styles (after Kolb)

It is suggested that different Learning Styles are suited to specific learning environments – whether the topic itself is abstract or practical in execution [15]. The programming task traverses these environments, depending on whether a student is trying to solve a problem (a symbolically complex environment), applying skills (behaviourally complex), understanding and identifying the relationship between concepts (perceptually complex). This might suggest that different learning styles come to the fore during the overall programming process. The relevance of the LSI to success has implications for teaching [1]. Each individual is tested through a series of questions and their current position within the learning cycle can be represented by their position within one of the quadrants as shown. This designates one of the four learning styles: Accomodator, Diverger, Assimilator or Converger.

Students who have had less prior experience of using a computer before they embark on a programming course are likely to be less confident in the initial sessions of a programming course, and consequently display anxiety characteristics. Closely related to anxiety is the issue of self-efficacy – an estimation of ones' ability to successfully perform target behaviours. Individuals who judge themselves capable of performing certain tasks are found to attempt and successfully execute them. This area of study has shown significant results [16], and a tool by which computer self-efficacy may be measured has been developed [11]. Evidence of prior computing experience should therefor be included in any study of a novice computing course.

3 The course and study

The class under study was a first year course, *Programming and Logical Methods*, delivered to incoming Humanities students as one of four subjects they had chosen to study as part of their honours BA degree. The objective of the course was to teach students the broad concepts of programming with associated laboratory work where they might practice the skills acquired using the BASIC programming language. While the context of this degree programme was slightly different than that for first year computer science students, aims were similar. In both cases, success is designated as a measure of the ability to specify, design, code and test a computing solution.

The course comprised two hours per week lectures, and two hours laboratory sessions over two semesters. Assessment was by final three-hour written examination (70% marks), and by 20 weekly assignments (30% marks).

110 students completed the course and sat the final examinations.

Data was gathered from student surveys and academic records. It was analysed using a statistical software package.

4 Results

The scores for all students were examined against predisposition factors of gender, prior experience, Learning Style, and previous academic performance in mathematics, science and languages.

4.1 Gender and prior experience

The majority of students entering the BA degree are female, and this was reflected in those studying IT. Of the 110 who completed the course and sat final examinations, there were 67 female and 43 males (61% female). This trend is contrary to the composition of most computer science classes.

An examination of the results shows no significant performance difference for male and female students (as shown in Figure 2). The mean score for the whole group was 42.2%, and the median 41.2%.

Males had a mean score of 39.7%, and females a mean score of 43.9%. While the females scored slightly higher, this does not represent a significant value. However, it does display the fact that females are equally capable of performing well in introductory programming.

GENDER	N	Mean score	Std. Dev.
Male	43	39.7	16.5
Female	67	43.9	16.2
Total	110	42.2	16.4
F-test		0.900	
t-test		0.193	

Figure 2. Mean exam score by gender

There is much evidence in the literature that for females, lack of early computing experience puts this group at a distinct disadvantage [7,2,14, 19].

Only 10 students in the whole group (male and female) reported having prior experience of programming, although most students (66%) had some previous interaction with computers, mainly with word processing or spreadsheet software packages.

GENDER	Experience	Mean score	N	Std. Dev.
Male	Limited	39.14	36	16.57
	Programmed	42.53	7	17.00
	Total male	39.69	43	16.48
Female	Limited	43.15	64	16.16
	Programmed	59.47	3	9.99
	Total female	43.88	67	16.24
Total	Limited	41.71	100	16.34
	Programmed	47.61	10	16.79
	Total	42.24	110	16.39

Figure 3. Mean score by gender and computer experience

In examining the results against prior programming experience, (Figure 3) females who have already programmed performed better than any other group. However, there were only three students in this group, too small a cohort to draw any significant conclusions.

Lack of previous experience was a disadvantage for both male and female students. These figures support previous research.

4.2 Learning Style

The 91 students who took the Kolb LSI test were categorised into the four learning types: Accomodator, Diverger, Assimilator and Converger. The largest group of students had Learning Style of Converger (37% overall). This group also performed best overall, as shown in Figure 4, although the results are not statistically significant.

Learning style	Mean	N	Std. Deviation
Converger	45.02	34	15.40
Diverger	40.68	15	14.49
Accommodator	43.88	11	20.12
Assimilator	43.52	31	14.08
Total	43.66	91	15.26

Figure 4. Mean score by learning style.

Those with Converger style combine abstract conceptualisation and active experimentation, and are deemed best at finding practical uses for ideas and theories. Their strengths are said to be in problem solving, decision making, deductive reasoning and defining problems. This combines many of the attributes which are required for successful programmers.

4.3 Prior academic performance

Most of the students (90%) had entered the course directly from secondary school, and had gained entry to college based on their results in the Irish Leaving Certificate examination. Students take six subjects in the Leaving Certificate – Mathematics, English, Irish, a modern language and two others. There are two levels of examination for each subject – Pass or Honours level. In calculating points for university entrance requirement, there is an equivalency table which allocates a number of points for each grade. This table was used to determine the academic attainment of each student. The correlations of the Leaving Certificate subjects of Mathematics, Science, English and languages are shown in Figure 5.

Examination score	Pearson Correlation
Mathematics	.353*
Science	.572*
English	.088
Foreign language	.119

*significant at the .01 level.

Figure 5. Pearson correlation factors for Leaving Certificate subjects

The Leaving Certificate Mathematics course is considered in particular to be proportionally more difficult at Honours level, and many students do not select to do the more complex course unless they are intending to study courses for which it is required. In all, only 29 of the 101 students who had sat the Leaving Certificate Mathematics exam had taken the Honours course, and only 2 had attained maximum (100) points. The maximum score for a student taking the Pass course is 60 points. The mean score in Leaving Certificate Mathematics was 53%.

There is a belief that the concepts which a student has to comprehend in order to master mathematics problems are similar to those for programming. Mathematics aptitude is thus often a pre-requisite for acceptance into computer science.

The results show a correlation of .353 between Mathematics points and programming examination score. This is a significant result at the .01 level. These results support the theories and research to date.

The students in this study had chosen to study Humanities rather than Science subjects for their degree, and thus many did not have a Leaving Certificate science subject on entering college. Three mathematical science subjects on offer for the Leaving Certificate were considered for analysis. These are Physics, Chemistry, and a more general subject named Physics and Chemistry. The best mark of any of these taken was considered, as no one subject would have given a sufficient number of students from which to draw conclusions.

Although there has been no empirical research to link science subjects and programming success, computing is often considered as a laboratory subject in the university and experience with other laboratory-based disciplines might be considered an advantage.

In all, 35 students had studied a science subject to Leaving Certificate level. Their mean score in science was 62%.

The results show a correlation of .572 between Science points and programming examination score. This is a significant result at the .01 level.

It is a requirement that students entering the Arts faculty will have a foreign language at Leaving Certificate. In all, 99 students reported marks for at least one of the languages French, German or Spanish. The best score obtained was used for analysis. The mean Leaving Certificate foreign language score was 69%.

There have been no direct studies comparing foreign language skills with programming skills. However, it might be considered that learning a programming language might have some basic similarities to learning any language. In the coding phase of programming, attention to construction and syntax might be considered similar to language grammar skills, and creative writing skills might be considered similar to developing innovative programming solutions.

The results show a correlation of .119 between foreign language points and programming examination score. This is not statistically significant. The links between foreign language and programming remain unproven.

5 Conclusions

While having no overwhelming message, the results of this study show a clear link between programming ability and existing aptitude in mathematics and science subjects. This could be that computing as a subject requires a structure and approach with which science students have some experience, and similar cognitive skills used in the study of mathematics. This supports the theory that we should take these subjects into consideration when selecting students for computer science courses, a practice evident in most schools of computing.

While no significant conclusions can be drawn from the study of Learning Styles, the predominance of a Convergent style among students who had selected to do the subject, and success of these students within the overall group, suggests that it is an area which invites further study.

Female students achieved equally high scores as their male counterparts, an outcome which contradicts some of the evidence found in previous studies. As might be expected, the few students with prior experience in programming outperformed those with limited experience.

References

[1] Bostrom,R.P. Olfman,L. & Sein,M.K. (1990), The Importance of Learning Style in End-User Training, MIS Quarterly March 1990.

[2] Camp T. (1997), The incredible shrinking pipeline. Communications ACM 40:10.

[3] Carbone, A. & Kaasboll, J.J. (1998), A survey of methods used to evaluate computer science teaching, ITiCSE conference1998.

[4] Evans, G.E. & Simkin, M.G. (1989), What best predicts computer proficiency? Communications ACM 32:11, pp.1322-1327.

[5] Fisher, A. Margolis, J. Miller F. (1997), Undergraduate women in computer science: experience, motivation and culture. SIGCSE conference 1997.

[6] Goold, A & Rimmer, R. (2000) Factors affecting performance in First Year Programming, ACM CIGCSE Bulletin, Vol. 32, pp. 39 - 43.

[7] Guinan, T. & Stephens, L. (1988), Factors affecting the achievement of high school students in beginning computer science courses, Journal of Computers in Mathematics and Science Teaching .

[8] Hostetler, T.R. (1983), Predicting student success in an introductory programming course, ACM SIGCSE Bulletin 15:3, pp. 40-49.

[9] Kolb, D. (1985) Leaning Style Inventory, Hay/McBer, Boston,Ma.

[10] Mc Quillan, H. & Bradley, M. R. (1999) Barriers for Women in Computing, University of Limerick.

[11] Murphy, C.A., Coover, D & Owen, S.V. (1989), Development and validation of the computer self-efficacy scale, Educational and Psychological Measurement 49, pp.893-899.

[12] Norton, D.A. Williamson, P.N. & Waker, P. (1990), Criteria by which pupils can be selected to offer computer studies in the Senior Secondary school, Proceedings IFIP world conference on Computers in Education 1990.

[13] Scragg, G. Smith, J. (1998), A study of barriers to women in undergraduate computer science, ACM SIGCSE conference 1998.

[14] Selby, R. Ryba, K. Young, A. (1998), Women in computing: what does the data show? ACM SIGCSE Bulletin 30:4.

[15] Sims,R.R. (1990),Adapting training to trainee learning styles, Journal of European industrial training 14:2, pp.17-22.

[16] Winslow, L. E. (1996), Programming pedagogy - a psychological overview, ACM SIGCSE Bulletin

[17] Woodhouse,D. (1983), Student selection for computing courses, Australian Computer Journal