

Why do Students take Programming Modules?

Paul Curzon
Middlesex University
Bounds Green Road
London N11 2NQ, UK
+44 181 362 6762

P.Curzon@mdx.ac.uk

Janet Rix

Middlesex University
Bounds Green Road
London N11 2NQ, UK
+44 181 362 6228

J.Rix@mdx.ac.uk

1. ABSTRACT

Programming jobs are increasingly scarce and yet an increasing number of students are taking IT and computing courses containing programming. It may therefore be that many will never need to write programs themselves. Instead the knowledge and skills may be primarily of indirect use in other IT areas. If being able to program is not the ultimate objective of students taking such modules, then the aims and objectives of curriculum developers may need to be changed accordingly. In this paper, we investigate the motivations that University students have for taking programming modules, and look at how those motivations change over time.

1.1 Keywords

Programming, changing motivations.

2. INTRODUCTION

Student motivation is of critical importance and can have a major impact on student's learning [1,4]. It is commonly assumed when preparing programming modules, that the main aim is to teach students the skill of programming so that they can get programming jobs. However, it was suggested at the 5th Annual Conference on the Teaching of Computing [2] that this is not what motivates students to take such courses. Programming jobs are comparatively rare. It is thus plausible that the majority of students taking programming courses will not actually use those skills. Instead knowledge of programming may just be a useful secondary skill for the IT professional to possess. A further

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITiCSE '98 Dublin, Ireland

© 1998 ACM 1-58113-000-7/98/0008... \$5.00

possibility is that the main use of programming modules is to teach more transferable skills such as general problem solving and logical thinking. However, if this aim is the true aim the way the subject is taught should be different.

In an attempt to shed light on the above argument we conducted a study on the students of three programming modules at Middlesex University. Our aim was not only to look at what the motivations of the students were, but also investigate how those motivations change as the students progress through University, taking more advanced programming modules. A more general study is also being conducted by Newstead et al [6,7]. It is in its early stages, however, and is concerned with the motivations of students over a wide range of disciplines, whereas we are only concerned here with the reasons for students taking programming modules.

Our study concerned three programming modules. The first was a first year, first semester introductory programming module on structured programming. It was based on the structured programming subset of C++. The second was a second semester follow-on module looking at more advanced programming constructs such as recursion and dynamic data structures. It again used the structured programming subset of C++. The third was a more advanced, year 2 module on general object oriented programming in C++. Students studying the latter had passed the former two modules in the previous year. All three modules contain a key skills strand. In the first year modules, this consists of a series of seminars on subjects such as giving presentations, group work and note taking. General problem solving is also included. All three modules are taken by a mixture of BSc Applied Computing, HND Computing and Joint honours degree students. A small number of Joint honours students drop programming after the first module. Others may drop programming after the second. All BSc and HND students must do all the programming modules. However, many weaker students failed the year and so did not progress to the second year.

3. METHODOLOGY

For the introductory module, the study was conducted in the first lecture in the students' first week at University. The first part of the lecture consisted of a brief outline of what would be taught on the module, namely an introduction to

programming suited both to complete beginners but also to teach good programming practice and underlying concepts to the more experienced programmers. The key skills strand of the module was also outlined. The students were then asked to write down the single most important reason for them doing the module. It was emphasised that there were no right answers, only the student's personal answer. After they had written down their main motivation a questionnaire about past experience and motivations was given out. This questionnaire asked for the student's name so was not anonymous, though students could omit giving their name. The questionnaire included a question asking how important it was to the individual to be able to program. It also asked them to copy on to the form the motivation they had just written down. They were then asked to choose three motivations for wanting to take and pass the module from a list of 10. These were:

- Because you think it will be easy
- Because you have been told you have to do it
- To learn how to program
- Because you wish to become a professional programmer
- Because you would like to write programs for your own use
- Because an understanding of programming will help in your future career
- To understand more about programs and programming
- To improve your general ability to solve problems
- To learn general study and transferable skills
- To get the module credits

We later categorised the original written primary motivation into one of the above areas.

The same survey was then carried out on the same group of students on the second module at the start of the following semester. Again, this was done in the first lecture. This was before the results from the first module were published. For the final object-oriented programming module, the survey was carried out in the last lecture. This was the very end of

the formal teaching they would receive on general programming skills. As all the surveys were carried out in the 1997/8 academic year, the students of the third module were different to those used for the first two.

The sample size (i.e. number of returns) for the first year, first semester group was 115, for the first year second semester group, 134 and for the second year group 42. The total class size (i.e. potential number of returns) for the first years was approximately 180, and for the second year group 114. Possible reasons for a non-return could have been non-attendance at the lecture where the survey was conducted, or the student declining to fill-in or hand in the form.

4. RESULTS

The results are given in Tables 1 and 2. Table 1 shows the percentage of students whose main motivation chosen fitted most closely into each of the categories. Table 2 gives the percentage that each category was chosen when the students were asked to pick three from the list.

4.1 Results from the Start of Semester 1

It is often suggested that students look for the easiest route to get a degree, and are not really interested in learning. If this were so, one would expect that a major motivation for doing the module would be to get the module credits or because the module was perceived to be easy. For example, this might be so for those who had done programming previously. Over a third claimed to have some programming experience, but the majority of these stated they were not confident even to write simple programs. A further possibility is that the main reason that students do the programming modules is because they are core modules: they must do the module to major in Applied Computing. If this were a major motivation one would expect students to indicate that they were doing it purely because they had to. Another reason students may wish to do the module might be to learn how to program for their own use, e.g., for games. In fact these reasons all barely registered, being chosen by only a few percent of the first year students when they first arrive (see Table 1).

Motivation	% giving it as main motivation		
	Semester 1: start	Semester 2: start	Semester 3: end
Because you think it will be easy	0	0	0
Because you have been told you have to do it	1	5	19
To learn how to program	23	33	29
Because you wish to become a professional programmer	20	15	2
Because you would like to write programs for your own use	0	0	0
Because an understanding of programming will help in your future career	33	27	17
To understand more about programs and programming	20	15	24
To improve your general ability to solve problems	1.5	1	0
To learn general study and transferable skills	0	0	0
To get the module credits	1.5	4	9

Table 1: Main Unprompted Motivation Given by Students (here classified into 1 of 10 areas)

Motivation	% of times chosen as a motivation		
	Semester 1: start	Semester 2: start	Semester 3: end
Because you think it will be easy	0	0	1
Because you have been told you have to do it	2	5	9
To learn how to program	21	33	16
Because you wish to become a professional programmer	14	15	8
Because you would like to write programs for your own use	4	0	8
Because an understanding of programming will help in your future career	24	27	27
To understand more about programs and programming	14	15	14
To improve your general ability to solve problems	12	1	8
To learn general study and transferable skills	4	0	1
To get the module credits	5	4	8

Table 2: The Three Main Motivations of Students Selected from a List of Choices

The data suggests that there are a variety of motivations involved. For new students the motivations are mainly career based ones, followed by a desire to learn. The most common first motivation was because programming skills were seen to be generally career enhancing (33%), but a significant proportion (20%) were doing it specifically for a career as a programmer. This would appear to refute the suggestion that becoming a professional programmer was not a significant motivating factor at least when students start University. Similar proportions gave a learning oriented motivation: either just to learn how to program or to understand more about programming. The other categories were barely mentioned. Other suggestions given by a notable proportion that did not fit one of our categories were "because it is interesting" and "because I need it for the academic subject I am actually interested in". The latter was mainly from students who were not majoring in Computing and taking the module as an elective.

When the students were given a choice of three categories the results were similar. However, interestingly "to improve your general ability to solve problems" gained a significant 12% of the vote. Thus problem solving was seen as a useful transferable skill to learn. This contrasted with "learning general transferable skills" which still only gained 4% of the vote. Thus being taught transferable skills in general was not seen as very important, but the specific skill of problem solving was. This was despite the very poor initial skill levels of the class at essay writing, presentations, etc.

4.2 Results from the Start of Semester 2

By the start of the second semester, the motivations of the students had started to change. Slightly more were doing it because they were told to or just to get the module credits. However, the only significant choices were still the learning and career oriented ones. There was, however, a trend away from career based motivations (dropping by 5-6%) towards learning based ones.

When the three main motivations were chosen, the trend away from career based motivations disappeared. The learning motivations still increased in popularity but at the

expense of transferable skills, which now barely registered. This suggests that the module had not fulfilled their expectations with respect to learning about problem solving in general.

4.3 Results from the End of Semester 3

The results for the second years showed some differences (see Table 1). The main motivation written was now "to learn how to program" (29%). "Understanding how to program" was similarly high at 24%. However, there was a significant fall in the number of students taking the course for career related reasons. Only 17% gave a general career reason and only 2% now gave a desire to be a professional programmer as the primary reason. Instead 19% were doing the module because they had to and 8% just to get the module credits. Thus the trends that had started in the first semester continued and even accelerated. In particular, strategic motivations were now significant. A large proportion of students now had no interest in programming per se. Very few still saw programming as their future career, and a reduced, though still sizeable proportion were doing the module primarily for other career based reasons.

When given the choice of three options from the list, the cynic's choice "because I was told to" dropped to only 9% of the vote. The general career option rose to 27% of the choices and the professional programming option rose to 9%. The two learning oriented choices fell significantly: learning to program falling to 16% and "understanding more about programming" falling to 14%. As with the first semester group's results there was a rise in the selection of problem solving (to 8%), with general transferable skills barely registering.

5. CONCLUSIONS

At least when they arrive, becoming a professional programmer is a major motivation for students to study programming. However, this motivation is lost as students progress through University, gaining a clearer idea of what programming involves and of what their career aspirations really are. Programming is still seen as useful career-wise, but only as a secondary skill. Learning general problem

solving skills, whilst not a primary motivation, is something that many see as a useful reason for taking a programming module. However, if such a skill is not obviously delivered, its importance disappears.

A certain amount of caution must be used when interpreting the above results, however. For example, the order of the motivations might have affected the choices. Also it is possible that the students did try to give what they perceived as the right answers. The introduction to the module given at the start of the lecture could have had an effect in this way. However, if this were so, one would expect transferable skills to score more highly as it had been indicated that they formed a significant part of the module. The fact that only the stronger students who passed both first year modules were allowed to take the second year module may also affect the results. If anything one would have thought that this would have increased the number intending to become professional programmers since these were the students that this was likely to be a possibility.

6. DISCUSSION

The students appear not to ultimately expect to become programmers or directly use their programming skills. This suggests that emphasis when designing such modules should not be focused just on an ability to write programs. Instead, if the intention is to equip students with useful and relevant skills, it should be primarily concerned with programming principles, logical thinking and problem solving.

Motivations are not fixed within students – educational settings can affect them: both positively and negatively [3]. Attempts to foster student motivation are evident in programming courses. There has been a move away from clean teaching oriented languages such as Pascal towards commercially relevant languages such as C++. One reason for this is that it is perceived to have motivational advantages: students “want” to learn C++ as that is what employer’s ask for. This would suggest that students will be more motivated to learn and thus will learn to program more effectively. “Messy details” of such languages have to be learnt if a student is to program commercially, so may as well be learnt from the start. However, the opposite effect could be occurring – the use of a language such as C++ could be one of the factors de-motivating the students due to the language being harder to learn. The details of the language could also be obscuring the principles that will actually be most useful for the students to understand.

7. FURTHER WORK

As a follow up, we intend to continue to track the motivations of a single group through their University career. This would overcome the problem of comparing two disparate groups. It would also be useful to do a follow-up survey to discover the kinds of jobs the student’s ultimately

obtained and whether they did directly need programming skills for those jobs. Differences between the first and second year students could have been due to other factors such as the different backgrounds, rather than due to a change as they progressed through University. This study was also done at a single University. It would be useful to conduct the same study at other institutions offering similar programmes. For example, initial motivations and subsequent changes may be different for students at further education colleges. Isroff and del Soldato [5], for example, found differences in motivations between students at a traditional University (UCL) and at a distance learning University (OU). It would also be useful to investigate if the language used does affect changes in motivations.

We intend to further analyse the data to investigate the differences between the different groups taking the modules – HND, BSc and Joint Honours students. For example, an initial analysis suggests that the biggest move towards “doing the module because you have to” in the second years was amongst HND students. Similarly, we intend to examine any links between changing motivations and results.

8. ACKNOWLEDGMENTS

We are grateful to Julie Marshall for her help in conducting this work.

9. REFERENCES

- [1] Brown, S., Armstrong, S., and Thompson, G. (eds) *Motivating Students*, Kogan Page, 1998.
- [2] Daughton, G., and Magee, P. *Proceedings of the 5th Annual Conference on the Teaching of Computing*, August 1997. Centre for Teaching Computing, Dublin City University.
- [3] Entwistle, N. *Motivation and Approaches to Learning: Motivating and Conceptions of Teaching*. Chapter 3 of *Motivating Students*, edited by Sally Brown, Steve Armstrong and Gail Thompson, pages 15-24, Kogan Page, 1998.
- [4] Farthing, D. W. (1997) *The three M’s of Open Learning: Medium, Material and Motivation*. In *Proceedings of the 5th Annual Conference on the Teaching of Computing*, edited by Goretta Daughton and Patricia Magee, August 1997. Centre for Teaching Computing, Dublin City University.
- [5] Isroff, K., and del Soldato, T. *Students’ motivation in Higher Education contexts*. Chapter 8 of *Motivating Students* edited by Sally Brown, Steve Armstrong and Gail Thompson, pages 73-82, Kogan Page, 1998.
- [6] Magee, R., Baldwin, A., Newstead, S., and Fullerton, H. *Age, Gender and Course Differences in Approaches to Studying in First-year Undergraduate Students*. Chapter 9 of *Motivating Students*, edited by Sally

Brown, Steve Armstrong and Gail Thompson, pages 83-92, Kogan Page, 1998.

[7] Newstead, S. Individual Differences in Student Motivation. Chapter 19 of *Motivating Students*, edited

by Sally Brown, Steve Armstrong and Gail Thompson, pages 189-199, Kogan Page, 1998.