

## HOW DO FIRST YEAR IT STUDENTS GO ABOUT LEARNING TO PROGRAM?

Research team: Ass. Prof. Christine Bruce, Dr John Hynd,  
Mr Michael Roggenkamp, Camille McMahon

Thank you for your participation in the project “Understanding students’ experiences of learning to program”. Several students were interviewed and the interviews analysed to discover the group experience. The following is meant to give you a taste of what was found.

This research, investigating variation in first year university students’ early experiences of learning to program, focussed particularly on revealing differences in experience. The analysis process revealed five different ways in which students go about learning to program in introductory university level units. These are portrayed in categories of description which capture the critical dimensions of what students learn as well as how they go about learning. Students may go about learning to program by:

- Following – where learning to program is experienced as ‘getting through’ the unit. When experiencing learning to program this way, the student may struggle to keep up with the set assignments. Where marks are to be gained, the student will focus on those tasks. Feedback is sought from teaching staff particularly to indicate whether the student is on the right track.

“I think there has to be some assurance to a person that if they follow the course, you know if they follow the tasks, that it gets somewhere...”(7:11)

- Coding – where learning to program is experienced as learning to code. Students going about learning to program this way see learning the syntax of the programming language as central to learning to program. This may involve rote learning. Students experiencing or going about learning to program this way may desire extra guidance from staff in terms of directing them to specific solutions and examples of code. That is, time taken to explore concepts and discover their own solutions, is considered wasted.

“...it’s like you’re trying to learn Russian and you’ve got to know the words you know....going over the lecture notes before the lecture and that, you know, (laughs) doesn’t make any difference, because you have to learn the syntax you know, and you can’t learn the syntax by just reading the chapter” (1:4)

- Understanding and integrating – where learning to program is experienced as learning to write a program through understanding and integrating concepts. When experiencing or going about learning to program this way the student sees understanding as integral to learning. They tend to seek understanding of a ‘bigger picture’ over the small tasks they are undertaking as part of the coursework. It is not enough to type in the code and ‘see if it works’, but rather the student seeks to understand what they have done in order to affect the particular outcome.

“Oh ok yeah, well it’s concepts. It really is ... I mean, there’s a whole side of things away from the keyboard ... the concepts of how ... Ok. It’s for want of a better word, I

think it's synergy – basically the sum of the parts [is] ... Greater than the actual parts involved ... you've got to learn the overall concept of the programming ..." (8:13)

- Problem solving – where learning to program is experienced as learning to do what it takes to solve a problem. When going about learning to program this way the student has the problem as the starting point and sets out to discover the means to solve that problem. Understanding is obtained through adopting a 'big picture' perspective, or trying to see the problem and the program as part of a broader context.

"Whereas if you can sit down and you know the individual components but you've got this larger problem to solve, well then you can go in and do it" (2:2)

- Participating or enculturation – where learning to program is experienced as discovering what it means to become a programmer. Learning to program is experienced as learning what it takes to be a part of the programming community. Understanding of what it means to learn to program encompasses the way that programmers think as well as what a programmer actually does. The previous focal elements of syntax, semantics and the logic of the programs are acknowledged in this category, but the meaning of what it means to learn to program extends to the actual programming community.

"... learning to program ... I guess it's just getting to grips with the thinking behind writing computer programs and then getting to grips with the code, the semantics of the language and ... the kind of logic behind it all. Just basically ... it's almost a different way of thinking, computer programming. You've got to be much more logical and you've got to basically write out every single step you're going to do, whereas if you say, give someone directions to go somewhere, you miss out bits ... they fill in the blanks. But with programming you've got to put every single exact step, and if you miss out anything, generally your program won't function the way you expect it to ... So I guess it is thinking like a programmer" (9:8)

This description of variation constitutes a framework within which one aspect of the teaching and learning of introductory programming, how students go about it, may be understood. The range of categories taken together suggests that some learning experiences may reinforce particular kinds of ways of going about learning to program; and therefore that teachers may wish to design learning experiences, tutorial activities or assignments that orient students towards the full range of possible ways of going about learning to program in introductory university units. Some teachers may wish to emphasise particular ways of going about learning from the set of categories described.

Your participation in this project has enabled us to understand how students experience learning to program, and thus enable teachers of programming to refine their approach to the task.

Thank you again.