



Monitoring and evaluating a redesigned first year programming course

Dianne Hagan, Judy Sheard and Ian Macdonald

Monash University

dianne.hagan@fcit.monash.edu.au

Abstract

A collaborative project between the Computing faculty and the Education faculty of Monash University was aimed at improving the teaching and learning of first year programming. After initial research had identified the problems, some improvements were attempted during 1996. One department was willing and able to make major changes to its subjects immediately. The sequence of two first year programming subjects was restructured and redesigned. The progress of the project was monitored via direct observation, email and the World Wide Web. An improvement in the percentage of students who achieved very good results marked the project as a success.

1 Background

In an effort to improve the quality of teaching and learning in the Faculty of Computing and Information Technology at Monash University, the Dean in 1994 invited the Education Faculty to share their expertise in teaching. The Education faculty believed that simply presenting a series of seminars on the latest educational theory was unlikely to produce any lasting or significant change, and suggested a collaborative project that would first identify the problems and then suggest solutions.

The first year programming subjects were chosen as the target area for a pilot project, because every Department in the Faculty teaches first year programming, although with different implementation languages and different modes of delivery.

In the past, programming was considered the most difficult and least interesting subject by most first year students in all Computing courses at Monash University. First year programming subjects had comparatively low pass rates, and a low percentage of students chose to continue with programming subjects afterwards.

Permission to make digital/hard copy of part or all this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. ITiCSE '97 Uppsala, Sweden
© 1997 ACM 0-89791-923-8/97/0006...\$3.50

The Departments of Software Development and Computer Science volunteered to take part in the project. An initial research project involving observations of lectures and tutorial classes by education experts, plus a series of detailed interviews of most staff members and a randomly selected group of students over a one semester period, produced a clear picture of the problems and a set of recommendations for solving them.

Some of these recommendations related to the training of tutorial staff in educational techniques, as most of them had no prior formal training in education and the research had shown clearly that tutorials, not lectures, were the students' major learning environment. A three day training course for tutors was developed, with a follow-up meeting every week or two during the semester. Both Departments adopted this practice, described in detail in [3]. Only the Department of Software Development was willing and able immediately to make major changes to the format, presentation style and content of its lectures and other classes. Its first year programming subject has about 400 students, of whom about 350 are in the daytime stream and the remainder attend evening classes.

2 Changes made to programming subjects

A team approach was instigated in order to foster "ownership" of the subject by the 18 tutors involved. As well as large quantities of email flowing between lecturers and tutors, the weekly meeting attended by an education expert specifically addressed teaching issues such as how to present various topics to the students, and how successful or unsuccessful the previous week's classes had been. In the Department of Software Development, there has always been a policy that the lecturer in a subject should also take at least one tutorial in that subject, in order to keep in touch with the students and to observe how they are coping with the material. The lecturers are therefore part of the tutoring team.

Each student had previously attended one two hour lecture and one two hour laboratory session each week. The lecture is attended by all students in either the daytime stream or the evening stream. For other classes, students are divided into groups of about 16, each with a tutor. For many years, students have been invited to ask questions of the lecturer and their tutors via email. They have availed themselves of this facility to a great extent, as it is often the easiest and fastest way of finding and gaining the attention of busy staff members regarding small problems.

In 1996, the two hour lecture was broken into two one hour lectures for the daytime students. Lectures were now seen as a way to introduce concepts and stimulate interest, but were no longer expected to induce deep understanding in the students. The major learning vehicles were discussion classes and laboratory sessions.

In addition to the two hour laboratory session, a one hour discussion class was introduced. It encouraged students to explore concepts introduced in the first lecture and to raise questions that could be answered in the second lecture. Students were required to discuss their ideas, and tutors were discouraged from giving answers and asking closed questions. Exercises incorporating educational techniques such as Predict-Observe-Explain, mimics, role playing and grids [1] were used in discussion classes. In first semester, these exercises were devised during the lecture preparation meeting, produced by one of the lecturers and given to the other tutors at the weekly meeting soon afterwards. In second semester, the tutors were given more autonomy and expected to produce their own materials for discussion classes.

In the laboratory session, students had the opportunity to write programs to cement their understanding of concepts and techniques. Collaboration between students was strongly encouraged, as the more students talk about the work, the more they understand.

Each lecture was discussed at length with education experts whose prior teaching experience was mainly with secondary school students. Several techniques that had been found valuable in that context were tried and evaluated, with varying degrees of success. The lecture preparation meeting focused on answers to questions such as "Do the students really need to know this? Do they need to know it *now*? How does it link to what they already know?". Different ways of presenting new topics were evaluated. Rather than delivering one lecture that covered all that students needed to know about a topic, an iterative approach was adopted whereby the topic was covered a little at a time, allowing students to become familiar with one aspect of the topic before introducing another aspect of it in a later week.

The education experts and the programming lecturers together produced a concept map of the subject content, and the education experts remarked that it was no wonder that students have trouble learning programming, as there are a very large number of concepts involved and the subject is extremely cumulative.

High quality Powerpoint slides were used to present all lectures, as were demonstrations of coding and executing programs using the environment available in the labs. These Powerpoint files and sample programs were available to students for downloading from the subject home page on the World Wide Web.

3 Use of the World Wide Web

The World Wide Web was used not only to disseminate all subject information including syllabus, lecture notes, staff timetables, exercises, assignment specifications, helpful hints, etc., but also to obtain feedback from students. Apart from the surveys described in the following section, a newsgroup was set

up to facilitate discussions. For technical (and other) reasons, this newsgroup was not used extensively.

A more popular facility was the anonymous feedback section. Anonymous comments and questions submitted via forms were answered by the lecturer or one of the other tutors almost immediately. This was well used in second semester, when students were familiar with the Web. Weaker students seemed more confident communicating with staff anonymously than contributing to a newsgroup or emailing tutors directly. At one point a student made some comments highly offensive to other students and staff; a filter program ensured that this did not happen again.

Anonymous feedback ranged from questions about why certain standards were set to complaints about the difficulty of the subject to compliments on its improvement. An example of some of the anonymous feedback received is shown below:

I would just like to thank all staff who were a part of this subject this semester. I have done this subject in the past and deferred because of the difficulty I experienced in this subject. In the past, the staff were hard to approach and the subject content itself was too much to take in during one semester. It seemed that this semester was much better organised and orientated at helping students understand the subject not just do it! The tutes assisted with the programming and the discussion group with the understanding of the programming. I felt comfortable with the work load and the content of the subject and felt that I gave it my best as it was not intimidating. Thankyou for the support, for the first time I actually ENJOYED programming and learnt something in the process. If the subject is organised the same way next semester, I'm sure future sft1102 students will feel the same. Thanks once again.

The use of email by students to communicate with tutors and the lecturer was greatly reduced.

Each tutorial group had a spot on the Web to display their work, in order to encourage collaborative learning. This also allowed the lecturer to see what different tutorial groups were doing. Students were invited to submit suggestions for examination questions via the Web, with a promise that the best three would be used on the examination paper.

The use of the World Wide Web is described in more detail in [2].

4 Monitoring of the project

Apart from anonymous comments, feedback was sought from all students by occasionally at the beginning of a week's laboratory session asking them to make radio button responses via the Web to questions about how they were coping with the subject and which components were most useful to them. An example of such a question is:

Which component of the subject do you find most valuable for your learning?

- Lectures
- Discussion classes

- *Lab sessions*

A Perl script stored the responses in a file and automatically summarized them.

A subset of students chosen at random were asked to respond at greater length via email to similar questions.

As a result of this monitoring, occasionally a lecture intended to introduce a new topic was cancelled and replaced by a revision lecture on a troublesome topic or a role play by tutors of the assignment specification in order to clarify it.

Education research assistants continued to observe discussion classes and laboratory sessions to find out whether the techniques espoused in the training course were actually being used. There was a clear improvement in good teaching practice between the first semester and the second semester, as measured by the amount and types of interaction between students and tutor. As no previous records had been kept, it was difficult to know how much improvement there had been between the previous year and the first semester.

Tutors were surveyed via email just after the initial training course and again at the end of semester to gauge their reactions to the training course, their estimate of its value and their impressions of the success of the subject. One of the questions and one response to it is:

> Do you think the subject was significantly different from other years? If so, how?

Yes, it was significantly different from other years from the POV of the obviously increased focus on quality of teaching throughout the subject - from what I have garnered from the students, they also see the differences.

Of course, this increase has meant an increase in workload during the average week for the tutors...

5 Success of the project

At the beginning of the project, the education experts predicted that the effects of the change in teaching style would probably not become apparent until second semester, as most of the tutors would take time to become comfortable with the new style. The student results (i.e., percentages of students achieving the various possible grades) for first semester were similar to results in previous years. In second semester, although the percentage of students failing or discontinuing the subject was similar to previous years, the percentage of students doing very well (i.e., achieving distinctions and high distinctions) rose from 31% to 40%.

Most of the tutors enjoyed participating in these subjects, even though they had to spend considerably more time and effort on them than on other subjects they were tutoring.

We feel that the project has been successful so far, and are looking forward to continuing it in 1997, further improving our subjects and extending the use of successful strategies to other subjects in the Faculty.

References

- 1 Baird, J.R. and Northfield, J.R. (Eds.) (1992) *Learning from the PEEL Experience*, Melbourne: Monash University.
- 2 Hagan, D.L. and Lowder, J. (1996) *Use of the World Wide Web in Introductory Computer Programming*, Proceedings of ASCILITE'96 conference, Adelaide.
- 3 Macdonald, I. and Hagan, D.L. (1996) *How Education Research Became Part of Teaching First Year Programming*, paper presented at AARE conference, Singapore.