Experience in Teaching Object-Oriented Concepts to First Year Students with Diverse Backgrounds

Experience in Teaching Object-Oriented Concepts to First Year Students with Diverse Backgrounds

Michael Blumenstein
School of Information Technology, Griffith University-Gold Coast Campus
PMB 50, Gold Coast Mail Centre, QLD 9726, Australia
E-mail: m.blumenstein@griffith.edu.au

Abstract

This paper describes the experiences in coordinating a first year programming course at Griffith University since Semester 1, 2000. In this time, the course structure and content have evolved to implement and evaluate an "objects-as-needed" approach to first year programming with the recent return to an "objects-early" approach. A variety of assessment strategies have also been employed in order to maximize student-learning outcomes. The success of the revised course has continuously been measured by evaluating student feedback and performance. Finally, a focus group-based strategy of evaluation was adopted to determine students' attitudes to the most recently implemented changes.

1. Introduction

In many institutions around the world a consensus has been reached to adopt Java as the language of choice for their first year programming course [1]. This by no means is a simple decision, as educators must ensure that the first programming course that students are exposed to will be a good foundation for future study. As Information Technology departments re-evaluate their programs, this issue presents itself anew, if and when the program structure is updated. Upon making the final decision to select Java, educators tend to be pre-occupied with the question of "how" to teach Java [2]. In the last three years at Griffith University, a number of issues have been tackled and revisited in terms of teaching a first language in the Bachelor of Information Technology (BIT).

One of the major issues that presented itself was the choice of methodology for teaching Java-based, object-oriented programming. In the last few years, a number of the favoured methodologies were employed, and evaluated. Another issue that has been prominent throughout the course's history is the diversity of the student population. This paper discusses the issues above and relays teaching experiences concerning program structure in the first year programming course at the School of Information Technology.

The remainder of this paper is divided into five Sections. Section two addresses the background and some challenges that were considered whilst evaluating the Introduction to Programming (ITP) course. Section three deals with the early evolution of the course, whereas Section four describes the various strategies that were used to further improve the course. Section five discusses the effectiveness of the newly implemented teaching strategies. Finally, Section six offers conclusions and provides some future directions for the ITP course.

2. Background and challenges

2.1 Student demographics

The ITP course at Griffith attracts a wide variety of students from different disciplines and backgrounds. For Multimedia and Information Technology students, ITP is a core course in the first year of their degree.

From past experience, the students enrolled in Semester 1 usually outperform the students in Semester 2 in terms of academic achievement. This seems to correlate with the fact that the majority of students enrolled in Semester 2 are Multimedia students or students from other disciplines. Through their own experiences Allen and Bluff [3] offer an explanation for the disparity between the two groups. They maintain that it arises due to the different expectations that each has when undertaking their respective programs. This evaluation has been confirmed whilst teaching the ITP course at Griffith University.

2.2 General challenges

In 2000 a major revision was undertaken to update the first year programming course (at the time called Application Design and Development) to improve its structure, delivery and finally its assessment strategies. One of the more important issues relating to delivery of a

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
0-7695-2108-8/04 $ 20.00 © 2004 IEEE

Java-based course, was the manner in which object-orientation was to be conveyed to students. Between 2000 and 2003, two main strategies were attempted and proved to be reasonably successful.

3. Evolution of the ITP course

This section details the evolution of ITP with reference to the revisions that took place during 2000-2001 (these have been detailed more extensively in [4] and will only be summarised in this section).

3.1 Teaching methodology

During the revision process, a decision was made to distance the course's content from the two radical approaches to teaching objects i.e. "objects-first" [5] and "structured programming-first" [6]. With the aid of the chosen textbook [7], an "objects-as needed" approach was adopted. This approach worked fairly well, as it enabled students to learn an object-oriented language without imposing excessive complexity from the outset.

3.2 Assessment

Throughout the life of the course it was deemed necessary to constantly evaluate and improve its assessment methodology. The approach of frequent student assessment was favoured however it was necessary to incorporate assessment pieces that could effectively evaluate students' individual performance. This was achieved through the introduction of a mid-semester and a final exam.

Issues relating to student motivation (in completing programming exercises) and low laboratory attendance in 2000 prompted a novel assessment strategy. It was decided that students would be asked to submit their weekly exercises for marks and would therefore be "obliged" to, at the very least, attempt the weekly exercises. The improvement in student learning outcomes based on this strategy was presented in [4].

4. Further revisions and the BIT review

4.1 Object-orientation issues

In a previous paper [4] it was reported that student performance and feedback from 2000 and 2001 coinciding with the period of the ITP curriculum and resources review proved to be quite successful. However, upon reflection and a closer evaluation of students' work, it became evident that there were still a number of students who were finding the concepts of object-orientation difficult to master. The objects-gently (or "as-needed") approach, although successful, had its drawbacks. Specifically, it was found that students were struggling at project time, attempting to master "cooperating" classes.

It was due to the above observations that it was decided to guide the ITP course through a further evolutionary step. In Semester 1 2002, the approach to teaching objects became less "gentle" and slightly more inclined to the "objects-first" approach [5]. It was possible to obtain good support from the 2nd edition of the Lambert & Osborne text [8] for this approach and hence it was selected for the 2002 academic year.

4.2 Teaching text file operations

Another curriculum change that took place in 2002 related to the manner in which reading and writing to text files was covered. In previous years, custom classes had been used in the course to abstract the difficulties of Java I/O and file operations away from students. Upon shifting to the new edition of the text, it was found that it contained its own custom classes for the purpose of making Java I/O operations simpler for students. It was decided to use these instead of those developed in-house. The main issue with this however, was that file operations were not handled by the textbook's classes and the standard stream classes were required to be taught instead. It was found that upon providing students with sufficient examples, they were able to master file operations employing Java standard classes with relative ease (to be discussed in a later section).

4.3 The BIT review

In 2001 a major review of the BIT program was commissioned and subsequently passed in 2002 [9]. One of the outcomes of the initial report required the identification of specific streams within the program including a programming stream. A sub-group of the School Committee produced a document for discussion [10] in order to propose the structure of the programming stream. The document's content placed considerable focus on the biggest point of contention: the choice of language for the first two programming courses. The document stated quite correctly that "…the choice of language should be dynamic, reflecting the current and projected programming landscape", nevertheless a decision was pending to choose between two preferred languages: Java and C++.

Various arguments were put forward for each particular language. In support of Java's case the following arguments were put forward [10]:

1. Simplicity

- Java does not contain some of the confusing features inherent in C++ such as pointers, operator overloading, multiple inheritance and templates
- Java handles storage management (i.e. automatic garbage collection), which gives it a big edge in introductory classes over C++

2. Robustness
3. Platform Independence
4. Portability
5. Easy GUI Programming
6. Easy transition to C++ and other languages
7. Low Cost

Upon discussion at length, a consensus was reached that enabled Java to retain its place as the first language of the BIT program, whereas C++ was chosen as the second language to be taught. The disparity between the languages taught in the first year of study in the BIT program raised some concerns. However, this issue will be evaluated in future reviews of the program.

It is interesting to note that one of the main reasons for selecting Java over C++ was its inherent simplicity for GUI programming. This prospect was especially attractive over the C++ option, which would be taught entirely in a console-based environment. It was noted that both Bachelor of Business and Multimedia students needed to be catered for and easy GUI programming was identified as one of the ways in which the students' experience could be enhanced whilst undertaking a first year programming course. It was pointed out at the BIT review that the new edition of the textbook [8] used in the Java-based ITP course already contained material for instruction of "easy" GUI programming which was pertinent to the diverse student population.

5. Discussion and evaluation

The following sections discuss some of the feedback obtained after the implementation of the various teaching methodologies and strategies mentioned in previous sections. The success of these strategies was measured in terms of learning outcomes, end of semester student feedback and focus group meeting feedback. The results of each are discussed in the ensuing sections.

5.1 Learning outcomes and feedback: 2002-2003

Student performance and course evaluations from 2000 and 2001 indicated that the newly instituted ITP curriculum proved to be quite successful [4]. In fact over the four semesters, the average student feedback pertaining to the course in general was 2.29 on a scale of one to five (one being the best). However, upon implementing the changes discussed above in 2002, it was necessary to perform follow up evaluations of student learning outcomes and feedback. Student performance over three semesters is shown below in Table 1 and student feedback is shown in Figure 1.

Table 1. Profile of Grades (Gr.) for ITP from Semester 1, 2002 up to and including Semester 1, 2003. The failure rate in brackets includes those students that did not submit the majority of assessment items.

Semester 1, 2002		Semester 2, 2002		Semester 1, 2003	
Gr.	%	Gr.	%	Gr.	%
HD	9.7	HD	11.0	HD	8.8
D	12.7	D	17.6	D	10.9
C	17.4	C	9.9	C	10.9
P	19.5	P	16.5	P	30.6
F	13.98 (25.8)	F	14.29 (25.3)	F	17.69 (27.9)

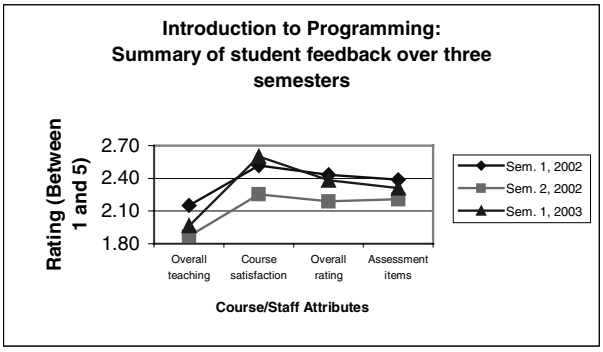

Figure 1. ITP: Student Feedback. The y-axis represents average student ratings on a scale between 1 and 5 (1 signifies 'Outstanding' and 5 signifies 'Very Poor')

As may be seen from the table above, over the three semesters listed, where minor changes to teaching methodology were implemented, a reasonably consistent learning outcome was obtained. Specifically, a good distribution of high achieving students can be seen over the three semesters. The failure rates are also consistent with those of 2000 and 2001, it may be noted that the failure rate (the figure that is not in brackets) represents the percentage of students that failed the course whilst attempting all assessment items. This figure is quite reasonable across the three semesters. It may be noted however that in Semester 1, 2003 the figures are slightly less consistent than in the previous two. This may be explained by the fact that in 2003, all courses across the university were condensed from fourteen to thirteen weeks. This necessitated some minor adjustments in the ITP course in terms of due dates for assessment items. As students had less time to complete some assessment items, some students found it difficult to cope with the workload in latter weeks.

Another issue was that 2003 saw the first time that the new BIT was implemented. This was the first time students were required to simultaneously undertake such courses as "Introduction to Logic and Algorithms" (ILA) as well as "Foundations of Computing" and finally "Fundamentals of Information Systems". These new courses, undertaken concurrently by all BIT students, may have had an impact on the student performance.

With regards to Figure 1, the student feedback results over three semesters are very encouraging. On a scale of one to five (where one is the best), the student's seemed pleased with the way the course was taught. It may be seen that the student satisfaction in Semester 2, 2002 was at its highest, however this seems to correlate well with the student performance for that semester and is inversely proportional to the number of non-IT students (only four out of the ninety enrolled that particular semester were Multimedia students).

Upon investigation of the number of students that failed or did not perform well in the course across the three semesters, it was found that over fifty percent of these did not sit the final exam. A majority of these were non-IT students undertaking a Multimedia or Commercial computing (Business) degree. At times, written student feedback from non-IT students conveyed a negative attitude towards programming in general. This seemed to correlate with the poor performance of these student groups.

5.2 Focus groups and ITP

It was anticipated that with the new modifications to the course curriculum in 2002 and the use of material for the rapid development of GUIs from the new textbook, the level of enjoyment and enthusiasm expressed by non-IT students might have increased. However, the feedback obtained at the end of each semester did not reflect this. At the same time, the feedback was not instructive as to the main problems that students came across in their assessment and the course in general.

With the advent of the course modifications described in previous sections, one of the initiatives considered important was to undertake a more thorough evaluation of student opinions and attitudes towards the course. The feedback obtained would be indicative of student attitudes towards specific assessment items and the new, less "gentle" approach to teaching objects as well as insights into the difficulties that students were experiencing with various aspects of the course.

The methodology chosen for evaluating the students was: The "focus group" approach [11]. In this approach, groups of students representative of the population are chosen to respond (in written form) and reflect on issues pertaining to the course at various intervals throughout the semester. Focus groups were convened from Semester 1, 2002 up till and including Semester 1, 2003. There were two sets of focus group "meetings" held during the course of a given semester. The first set occurred during laboratory time in either Weeks 6 or 7. This coincided with the completion of lectures dealing with the more assertive object-related material. The second set took place in Weeks 12 or 13 to determine how students were coping with advanced O-O concepts after being exposed to objects early on in the semester. In Semester 1, 2002 three labs, consisting of approximately 20 students each, were randomly chosen. In later semesters, two labs were used for the evaluation.

Ideally, the number of IT students and those from other disciplines should have had a reasonable representation in each lab. However, it must be noted that in some semesters there was an imbalance with regards to student numbers overall and this was reflected in the composition of particular laboratories.

For each focus group meeting, students were provided with two sets of questions. The first set related to either the in-class lab assessment or the project due in Week 6 and Week 13 respectively. These would gauge any problems students were having with particular aspects of the assessment. The second set of questions related to general aspects of the course and are reproduced below:

1. What are your current feelings towards the course?
2. What do you <u>like</u> about the course so far?
3. What do you <u>dislike</u> about the course?
4. What would you change about the course?

Upon obtaining feedback during each semester, the comments provided by students were reflected upon. Where patterns would emerge in the comments, i.e. difficulties experienced by a majority of students, appropriate adjustments were made in the ensuing semester in an attempt to increase learning outcomes. An analysis of the important aspects of the feedback is presented in the next section.

5.3 Analysis of focus group feedback

Upon analysis of the focus group feedback, some interesting patterns emerged. The focus groups that took place in Weeks 6 and 7 over the three semesters were undertaken to gauge whether the introduction of objects earlier on in the course caused problems for students. Surprisingly, in the survey over three semesters (109 respondents), only a single student expressed concern over the difficulties of dealing with objects. However, as may be seen in Figure 2, whilst completing their final projects in Weeks 13 and 14, students expressed the difficulties they were having with specific aspects of the course. In Semester 1, 2002, the big surprise was that

nearly 40% of students surveyed had problems with understanding arrays. However, it was found that approximately 20% of students had problems relating to object-oriented issues. Following this feedback, the course was further adjusted by improving the coverage of the above-mentioned topics in lectures and laboratory exercises. This seemed to assist in dropping these percentages over the ensuing two semesters.

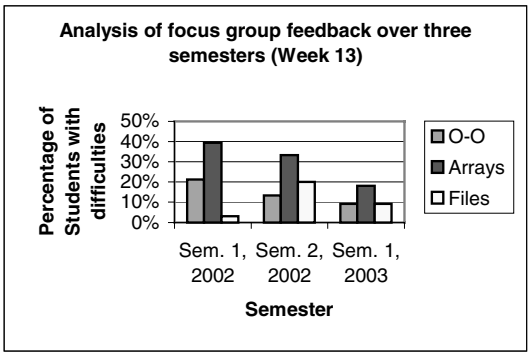

Figure 2. An analysis of the percentage of students facing difficulties in various aspects of the course

Aside from the above-mentioned difficulties experienced by students, other feedback about the assessment and course in general was quite positive. It was observed however that the first set of students surveyed (Semester 1, 2002), contained the largest number of multimedia students. In both Weeks 6 and 12, 66.7% of these students proclaimed that they did not need programming for their degree nor did they enjoy programming. These comments corroborated the results and other feedback from this student group. Unfortunately, it was not possible to report the responses of Multimedia students in ensuing semesters, as the number of Multimedia representatives surveyed was marginal.

6. Conclusions and the future of ITP

This paper has described the evolution of the ITP course at Griffith University, focusing specifically on the most recent modifications to course structure. Following the implementation of a more assertive approach to teaching object-orientation, the learning outcomes of students along with student feedback were measured over three semesters. It was observed that the learning outcomes and feedback over this period were encouraging and compared well to previous semesters, however some problem areas were still evident.

A focus group study was initiated over three semesters to obtain detailed information about student problems and perceptions of the course. Written feedback has suggested that students coped well with the new assertive approach to teaching object-oriented programming. It was also observed that feedback regarding areas of weakness in the course became more positive in later semesters, which may be attributed to further modifications to the teaching methodology.

In future, strategies for increasing the learning outcomes of Multimedia students and other non-IT students shall be investigated. This may include the addition of more relevant exercises for the above-mentioned student groups in addition to alternate assessment items. To benefit all students and staff in the future, an automated system for marking assessment items is currently being developed. Finally, software will also be developed for the creation of SDCs to be used by students in ITP.

7. References

[1] M. Guzdial and E. Soloway, "Teaching the Nintendo Generation to Program", *Communications of the ACM,* Vol. 45, No. 4, (2002), pp. 17-21.

[2] C. Wallace, P. Martin, and B. Lang, "Not Whether Java but how Java", *CTI Computing - Monitor 8, Java in the Computing Curriculum*, (1997), http://www.ulst.ac.uk/cticomp/not.html (downloaded 20/10/03).

[3] R. K. Allen, and K. Bluff, "Jumping into Java: Object-Oriented Software Development for the Masses", *ACSE '98*, (1998), pp. 165-172.

[4] M. Blumenstein, "Strategies for Improving a Java-based, First Year Programming Course", in *Proceedings of the International Conference on Computers in Education,* (2002), pp. 1095-1099.

[5] M. Kölling, and J. Rosenburg, "Guidelines for Teaching Object Orientation with Java", *ITiCSE 2001*, (2001), pp. 33-36.

[6] J. Gibbons, "Structured programming in Java", *CTI Computing - Monitor 9, Java in the Computing Curriculum II*, (Spring 1998), http://www.ulst.ac.uk/cticomp/gibbons.html (downloaded 20/10/03).

[7] K. A. Lambert, and M. Osborne, *JAVA: A Framework for Programming and Problem Solving* (1st Edition), Brooks/Cole Publishing Company, Pacific Grove CA, 1999.

[8] K. A. Lambert, and M. Osborne, *JAVA: A Framework for Programming and Problem Solving* (2nd Edition), Brooks/Cole Publishing Company, Pacific Grove CA, 2002.

[9] School of Information Technology, "Griffith University, Submission For A Major Change To The Bachelor Of Information Technology Program", *Internal Report*, (2002).

[10] M. Blumenstein, L. Della, W. Pullan, and J. Thornton, "BIT Review 2002 – Programming Options", *Internal Report*, School of Information Technology, (2002).

[11] D. Goodrum, M. Hackling, and L. Rennie, "The status and quality of teaching and learning of science in Australian schools", *A Research Report prepared for the Department of Education, Training and Youth Affairs*, (2001).