2002

Initial Experiences Gained and Initiatives employed in the Teaching of Java Programming in the Institute of Technology Tallaght

Eamonn Hyland
Technological University Dublin, eamonn.hyland@tudublin.ie

Gary Clynch
Technological University Dublin, gary.clynch@tudublin.ie

Recommended Citation

Hyland, E. and Clynch, G. Initial experiences gained and initiatives employed in the teaching of Java programming in the Institute of Technology Tallaght. Proceeding PPPJ '02/IRE '02 Proceedings of the inaugural conference on the Principles and Practice of programming, 2002.

Initial experiences gained and initiatives employed in the teaching of Java programming in the Institute of Technology Tallaght

Eamonn Hyland, Gary Clynch
Department of Computing,
Institute of Technology Tallaght,
Dublin 24,
Ireland.

eamonn.hyland@it-tallaght.ie
gary.clynch@it-tallaght.ie

Abstract

This paper describes recent experiences gained and initiatives employed in the teaching of Java programming to first and second year students in the Institute of Technology Tallaght from September 2000 to March 2002. It outlines some e-learning, technological and pedagogical initiatives that were undertaken within the department and the resultant preliminary outcomes. The outcomes have been determined after detailed analysis of the results of a survey which was commissioned to determine the effectiveness of the initiatives and also to identify those parts of the Java language which were causing students particular difficulty. The students surveyed are currently completing their fourth and final Java programming module. The Java programming modules make up the software development stream of a 2-year National Certificate in Computing (Information Systems). The typical profile of a first year computing student in IT Tallaght is someone who has achieved a reasonable but not exceptional level of academic attainment in the Leaving Certificate and who has had little or no previous exposure to software development. The design of the Java programming modules, while geared towards novices, is ambitious in an institutional learning context.

1. Introduction

This paper describes recent experiences gained and initiatives employed in the teaching of Java programming to first and second year students in IT Tallaght from September 2000 to March 2002 as part of a National Certificate in Computing (Information Systems). The paper presents the results of a survey conducted amongst the student body to determine the effectiveness of specific pedagogical initiatives undertaken with the aim of improving the learning experiences of students, and to determine those concepts in Java that students find most difficult to understand. The research was designed to be the first step in a series of surveys which will be conducted over the coming years. The study was specifically undertaken without any reference to published research in this area. The ultimate aim of the research will be to quantitatively measure the effectiveness of various pedagogical initiatives employed in the teaching of Java in the department and identify exactly those areas of Java that students have most difficulty with.

The structure of the paper is as follows. Section 2 gives a brief introduction to courses offered by the Department of Computing in IT Tallaght. Section 3 describes

some pedagogical initiatives undertaken with respect to the teaching of Java with the intention of enhancing the learning experiences of students. Section 4 provides the rationale for the survey and the survey design. Section 5 provides an analysis of the survey results. Section 6 provides conclusions.

2. Introduction to the Computing Department, IT Tallaght

The Department of Computing, IT Tallaght, offers a 4-year programme leading to a B.Sc. in Computing (Information Systems). The main thrust of the programme is the application of information technology to the solution of *business problems*. The primary objective of the programme is the production of professional information systems developers. The flexible structure of the programme provides students with the option of leaving after 2 years with a National Certificate in Computing, after 3 years with a National Diploma or after 4 years with a B.Sc. in Computing. The programme is designed to prepare students for entry into a range of systems development careers by equipping them with the requisite technical, business and interpersonal skills to meet the requirements of the software development industry.

3. Pedagogical Initiatives

When the software development stream of the National Certificate course underwent a major review in 2000, one outcome of that review was that Java programming would be taught from the start of the course for a period of 2 years full-time. New software development modules were developed (based on Java) and as they were being delivered for the first time, consideration was given to other pedagogical models that could be used to improve or add value to the software development learning experiences of students. Up until recently, lecturers in the department traditionally delivered software development lectures to students in lecture theatre settings using slides and a whiteboard. These lectures were then further supplemented with practical sessions in the computing laboratories. Continuous assessments were normally distributed on paper (either one-off in-lab assessments or projects over a period of time) and students completed and submitted these assessments with associated paper documentation. Although this method of lectures and assessment worked well, it was argued that this pedagogical model of teaching software development could be improved. The potential of e-learning, specifically the World-Wide-Web, multimedia and course management systems were considered to see if these technologies could be successfully exploited within the department to provide an ‘added-value’ dimension to the traditional teaching model.

Within this context, the *first* Java teaching initiative was implemented, which was the development of a departmental e-learning Intranet resource (Compweb i.e. http://compweb.it-tallaght.ie). Its primary purpose was to act as a repository of course materials for both staff and students involved with the Java modules. This resource was implemented prior to the start of the semester 1 module in September 2000 and was available on all workstations in the computing laboratories. The course notes were published in web format using standard hypertext navigational features. Compweb has been used to deliver not just course notes, but also course schedules, laboratory worksheets and solutions, sample programs, continuous assessments and solutions, continuous assessment results, and previous examination papers. It has since been expanded to include most modules within the B.Sc. in Computing.

The *second* teaching initiative included the introduction of a live coding approach in lectures. This approach involves the lecturer using a laptop and projector to demonstrate sample code, which supplements and illustrates the key points covered in the lectures. The lecturer goes through sample problems and solutions in real-time with the students.

The *third* teaching initiative was the commissioning of a special teaching laboratory equipped with an interactive whiteboard (smartboard) to further enhance the live coding/teaching approach. The smartboard is an interactive whiteboard coupled with a high-beam multimedia projector. The projector can be used to project a screen image from the instructor's or student's workstation onto the smartboard. The instructor can use special colour pens to write over the image on the smartboard as he/she would do with a traditional whiteboard. In addition to projecting images onto the smartboard and writing over them with special markers, synchronisation software enables the instructor to project the image currently on the smartboard onto the screen of every workstation in the laboratory. The instructor can therefore project the screen image of his/her workstation, or indeed the screen image of any student's workstation, onto the smartboard and also onto the screen of every workstation in the laboratory. This laboratory has been received so positively by staff and students that all software development classes are now timetabled there.

4. Survey Rationale and Design

To determine the effectiveness of these various initiatives and of the department's performance in teaching Java in general, a qualitative research study was undertaken. All of the 90 semester 4 (second year) students were surveyed. To conduct this research, the instrument considered most appropriate to the objectives and resources of the research was a questionnaire. The questionnaire was designed after a detailed analysis of the research objectives. Questions were developed to capture data to meet those objectives.

The students have completed 3 semesters of Java to date and they are currently covering GUI building, applets, and database and file I/O in their fourth semester. The topics covered in Semester 1 were introducing Java and the Java platform, variables and types, conditional statements, and looping statements. The topics covered in Semester 2 were arrays, objects and classes, and inheritance. The topics covered in Semester 3 were polymorphism, packages, exceptions, threads, and advanced language features. Programming style and practice were covered throughout the three semesters.

The questionnaire was designed to ascertain facts about *how* students rated the Java programming learning experience over the previous 3 semesters. Specifically the focus was to determine what difficulties students had with parts of the Java language, what parts of the language they found easy, and also how effective the extra teaching initiatives may have been.

The lecturers delivered the questionnaires to the students personally in a supervised classroom. Students were allocated class time to fill them in and were asked to treat the questionnaire seriously. 63 questionnaires were returned giving a response rate of approximately 70% of all second year students.

The questions were designed to accept a mixture of open ended and check box responses. The open-ended data was analyzed and interpreted with the detailed check box data and grouped under a number of overall headings.

5. Survey Results and Analysis

The purpose of the analysis is *to highlight any strong trends*, either positive or negative, which can help educators within the department to deliver the modules with added pedagogical value. Trends of interest include all aspects of course delivery, i.e. how Java is taught, facilities, course scheduling, course notes, worksheets, continuous assessments, exams etc. The purpose is also to discover those concepts in the Java language with which students have particular difficulty. The design of the survey was primarily qualitative in nature, though some numerical data has been produced which has been subjected to a quantitative analysis.

The first group of questions on the questionnaire inquired into the students' perception of the Java language over the 3 semesters in terms of which were the 'most 'difficult' topics and which were the 'easiest' topics to understand. Specifically the students were asked the following two questions:

> "*Please select the 3 most difficult topics to understand (in no particular order)"*
>
> *"Please select the 3 easiest topics to understand (in no particular order)"*

There were asked to pick from the list of topics for semester 1 to semester 3 inclusive as described in section 4 above. An analysis was performed where the number of times each topic was selected was counted, and divided by the total number of selections made. As regards the most difficult topics, *arrays* emerged as the most difficult topic to learn with 28% of total selections (Table 1). The next most difficult topic identified was *threads* at 20%. *Polymorphism* came next at 14%.

The expectation of the instructors was that *polymorphism* would prove to be an extremely difficult concept for students to grasp. The survey indicates that students did have a difficulty with polymorphism but that *arrays* were selected exactly twice as often as *polymorphism*. It is possible that the extra time and effort devoted to the teaching of *polymorphism*, in anticipation of perceived student difficulties, resulted in this positive outcome. It may also be the case that the instructional materials used to teach arrays were weak, that enough time was not devoted to the topic, or that comprehending complex data structures such as arrays is intrinsically a problem for students. Further analysis is required to determine exactly whether any of these, or some combination of these, is the case. The exercise will be repeated again to see if a firm trend emerges.

Table 1: Those topics that were considered 'most difficult'

Arrays	28%
Threads	20%
Polymorphism	14%
Looping Statements	10%
Exceptions	9%
Inheritance	6%
Objects and Classes	4%
Advanced Language Features	3%
Others combined	6%

As regards the easiest topics *inheritance* emerged as the easiest topic to understand with 18% of the total selections (see Table 2). This may be due to the fact that inheritance is a very natural concept that students can relate to, though further investigation is warranted before a firm conclusion could be reached in this regard. *Packages* and unexpectedly *conditional statements* came next at 16% and 15% respectively.

Table 2: Those topics that were considered 'easiest'

Inheritance	18%
Packages	16%
Conditional statements	15%
Exceptions	10%
Variables and Types	10%
Objects and Classes	8%
Looping statements	7%
Polymorphism	5%
Threads	4%
Others combined	7%

Another group of questions inquired into the students' perception of what were the best experiences and the main obstacles in learning Java programming in IT Tallaght. 45% praised the quality of the course notes and indicated that this was one of the best parts of the experience. Some interesting and related comments were their approval of the overall course structure and how they liked the way the course had been broken down into learnable component areas. One third of the sample indicated that they found Java very difficult and that this was the biggest obstacle they faced. Lack of access to facilities (i.e. computers during peak hours) also emerged as an obstacle for 37% of the sample. The other obstacle of mention that recurred was the unavailability of the Compweb Intranet resource outside the college. Finally some interesting comments emerged in relation to room facilities such as making the environment of the laboratories more attractive with the inclusion of plants etc.

The next group of questions inquired into the students' perception of how useful the three teaching initiatives were i.e. Compweb, utilising the laptops in the classroom to demonstrate live coding and the use of the special teaching laboratory with the interactive whiteboard (smartboard) to do the same. Specifically the questions asked were as follows:

"*How useful was the Compweb resource"*
"*How useful was the Live Coding approach"*
"*How useful was the Smartboard"*

In each case the possible answers were "Very useful", "Useful" or "Not useful at all". A breakdown of the answers to these questions is shown in Tables 3, 4 and 5 below. Overall all 3 initiatives were received very positively. The students have experience of more traditional types of teaching in other modules e.g. whiteboards, markers, slides, workshops etc. but nothing of this nature in software development. Of the 3 initiatives the Intranet resource Compweb received the highest "very useful" rating

with 97%, followed by the Smartboard with 74% and lastly the "live coding" approach with 67%. The Compweb resource facilitates self study by the students and this may have resulted in it being given the highest rating by the students.

Table 3: The Compweb Intranet resource

Very useful	97%
Useful	3%
Not useful at all	0%

Table 4: The "Live Coding" approach

Very useful	67%
Useful	30%
Not useful at all	3%

Table 5: The Smartboard resource

Very useful	74%
Useful	21%
Not useful at all	5%

6. Conclusions

The aim of the survey was to get some preliminary information about the effectiveness or otherwise of the pedagogical initiatives undertaken and to pinpoint specific problem areas within the Java language which could be targeted for improved and/or more focused teaching effort. No firm conclusions can be made at this stage based on the research results to date. A more quantitative analysis will be performed in the future before firm conclusions can be determined. Also no firm trends can be determined from the analysis of a survey of just one group of students at one point in time. It is intended that a new questionnaire will be designed and delivered at three separate time intervals in the future which will facilitate a more quantitative analysis.

A number of tentative conclusions have been reached as a result of this initial research. First of all, all three initiatives have been deemed to have educational merit as they have enriched the learning experiences of the students. The intention is that they will be further expanded in the future. The e-learning resource Compweb will be developed further and students will be given off campus access to it. Consideration is also being given to equipping all other computing laboratories with Smartboard technology.

Another conclusion is that further investigation is warranted into the whole topic of arrays. The instructional materials for arrays are to be reviewed and consideration given to the use of multimedia courseware to help convey pictorially the concepts of arrays, particularly multi-dimensional arrays. Also under investigation is the use of e-learning and course management software to support the Java stream.