TEACHING JAVA IN AN ON-LINE DEGREE PROGRAMME: A CASE STUDY

Marie Devlin, Frans Coenen, Paul Leng

Department of Computer Science,
The University of Liverpool,
Liverpool, L69 3BX
{marie,frans,phl\}@csc.liv.ac.uk}

Abstract

In this paper we describe our initial experiences arising from the delivery of a Java programming module to MSc students who are being taught entirely through the medium of the Internet. The mode of teaching is essentially seminar-based, with a heavy emphasis on the moderated discussion of module topics and coursework in a Virtual Classroom. This approach leads to a high degree of interaction between instructor and students, and, especially between students in the class, who collaborate to create a mutually supportive learning environment. We here outline the background and structure of this programme, and discuss way in which classroom discussion facilitates the teaching and learning of Java.

1. INTRODUCTION

In this paper we present some of our experiences concerning the delivery of Java as part of an online degree programme. The programme in question is an MSc in Information Technology offered by the University of Liverpool in partnership with a commercial organisation, K.I.T. eLearning of Rotterdam. The Java module that forms part of this programme, "Object-oriented Programming in Java" is of a kind that may be found in many degree programmes. What distinguishes it from others is that it is (to the knowledge of the authors) the only such module whose mode of delivery is entirely online. As such we consider it worthwhile to report on our experiences concerning the online delivery and assessment of a Java module presented as part of a taught degree programme. We also believe that our experiences may have some wider relevance with respect to: (a) the teaching of Java and (b) on-line delivery mechanisms.

The paper is organised as follows. In Section 2 we give a brief overview of the structure of the degree programme of which this module is a part. In Section 3 we present a short review of the module and its structure. The mode of delivery and assessment are described in Section 4, including, in Section 4.1, some comments on the issue of plagiarism. In Section 5 we present a discussion of our practical experience in delivering this module, and some conclusions are presented in Section 6.

2. The MSc in I.T.

The online MSc in I.T. offered by the University of Liverpool is delivered as a partnership between the University, represented by the Department of Computer Science, and K.I.T. eLearning of Rotterdam. In this partnership, the University retains full control over all academic aspects of the programme, while the commercial partner looks after issues of recruitment, marketing, and administration. The degree programme itself mirrors, as far as is possible, the academic structure of established programmes. Thus, the (online) MSc in I.T. comprises eight taught modules, each of 15 CAT units, followed by a dissertation project, and in this respect follows closely the model used for many conventional MSc programmes, including the one used in the parent University of Liverpool Department. The programme differs significantly from the 'on-ground'

MSc offered at Liverpool, however, in that it is targeted more specifically at working professionals in the I.T. industry who already have significant practical experience and sophisticated understanding of the field. Some implications of this are discussed in section 5. A fuller description of the programme and its background may be found in [1].

The distinctive feature of the programme is, of course, the module delivery mechanism. Each taught module, in general, is delivered entirely online over the Internet, over a period of eight weeks. For this purpose, the year is divided into five periods of ten weeks, allowing for two-week vacation periods between each module. If, as is expected to be the norm, a student pursues only one module at a time, this schedule would enable the full programme, including the final dissertation project, to be completed in about two years, although this can be extended to allow for longer vacation periods.

Module delivery involves the use of proprietary software to support a Virtual Classroom [2] for each module, within which a maximum of 20 students are guided by the module instructor. The software, the SoftArc FirstClassTM system, has been widely used for online teaching [3]. The mode of communication is asynchronous, necessarily, given the requirement to enable students to fit the demands of the course into their work schedule, and the additional problems of catering for an international community of staff and students who may be working in several different time zones. Associated with each virtual classroom is a set of mail folders to which students, teachers, and others involved in the course administration have access. Essentially, module delivery, and all other teaching interactions, take place via these folders. This deliberately low-technology approach makes the programme available to any student who has use of a PC with Internet access.

The teaching approach is essentially constructivist [4,5] with a heavy emphasis on discussion of module topics in the virtual classroom; further details are given in section 4. As part of the quality assurance regime for the programme, the delivery of each module is 'monitored' by a member of staff at the University's dedicated e-Learning Unit.

3. MODULE STRUCTURE

The Java module under discussion here is a standard introductory module, aimed at students without prior knowledge of the language, and is of a kind that may be found in many conventional degree programmes; its distinguishing feature is its on-line mode of delivery and assessment. However, prior to discussing the novel features of this model it is appropriate to give a few details concerning the nature of the Java module here.

The module comprises eight tutorials, one per week, as follows:

1. **Introduction:** Introduction to the Java programming language, the principles of Object-Oriented Programming (OOP), first Java program.
2. **Data:** Categories of data, data types, I/O and "wrapper" classes, arithmetic and the Math class, type conversion.
3. **Classes:** Method calls, inheritance and class hierarchies, abstract classes and interfaces.
4. **Selection:** "if-else" and "switch" and menu interfaces using switch statements.
5. **Iteration:** For loops, while loops and do-while loops; the break statement, nested loops, recursion.
6. **Data structures:** Arrays, array processing, unconstrained arrays, strings, arrays of objects.
7. **GUIs:** Introduction to GUIs, components, layout managers, graphics.
8. **Applets:** Introduction to applets, the Java Applets package, Applet parameters, GUI features and embedded URLs.

Each tutorial is accompanied by a set of discussion exercises to be "posted" during the week. Towards the end of each week each student is required to post a critique of at least one other student's response/answer to the discussion questions. The aim of the latter is to enable students to demonstrate evidence of the higher level of understanding commensurate with the higher degree under consideration here. Each tutorial also includes guided reading and private study questions (i.e. work to be done in the student's own time and not to be posted in the Virtual Classroom), and a practical exercise which will be assessed by the tutor. Each K.I.T. module has an adopted course text on which the "guided reading" is based. (Note: at present we have adopted ``Schaum's Outlines: Programming with Java'' [6]; it is easy to read, inexpensive and almost exactly matches the structure and material of our Java module).

4. DELIVERY AND ASSESSMENT

The Java module, like other modules in the MSC programme, is taught in a bounded-interactive manner and uses a combination of classroom-moderated discussions, individual hand-in assignments and individual and programming projects. There is no final exam and assessment is on-going and based on student's individual and group contributions.

At the beginning of each new class week, the instructor posts teaching materials and assignments for the students in the class. The instructor directs the students to the required reading sections in each chapter of the chosen text and outlines the major topics by posting lectures to the relevant **Seminar** folder. Students are encouraged to read other sections in the text and also to look at relevant sites on the WWW. After reading students can test themselves by solving problems from the chosen text or additional problems and participating in the discussion questions that are posted to the Virtual Classroom by the instructor.

The discussion questions are also posted to the **Seminar** folder and are a major part of the learning that takes place. They allow students to learn from each other, discuss current developments in the area and experience differing points of view. They provide the student with a feeling of 'involvement' that is sometimes assumed to be absent from the online environment. During the week the instructor gives feedback on and moderates the learning of students, monitoring their participation and providing them with weekly interim grades. For each week, the instructor records the different grades each student has received on the various assignments for the week, the attendance of the student and any special remarks. This report is posted to the **correspondence folder** of the class, so that all interested parties, (and only they) will be able to view the report. The ongoing assessment philosophy of K.I.T. has a lot of advantages in that we are able to keep track of students and recognise earlier than perhaps we would 'on the ground' when a student is having problems. This helps because online students are more likely to disappear or drop out if minor problems are allowed to escalate into big ones.

Each week the instructor monitors the discussions, answering questions that the students have regarding the material or the issues discussed, directing the discussions and highlighting the main issues covered in the lecture material and relevant text. At the end of each week, students are sent an assessment of their performance and are graded on their performance in discussions, the quality of their hand-in assessments and their attendance. The instructor also includes detailed individual feedback and provides pointers for improving performance etc.

4.1 Plagiarism

Plagiarism is, of course, an important issue with respect to all distance learning programmes; in the case of online programmes the opportunity appears greater because much of the material is in soft form and thus can easily be copied, transferred, etc.. In the teaching context we have described, however, we believe these problems are more apparent than real. The key point here is the role played by discussion in the Virtual Classroom. Because participation in discussion is an assessed requirement for students enrolled for a module it provides a means of monitoring their effective involvement, and assists in preventing impersonation and plagiarism. Involvement in the programme demands a thoughtful, personal contribution from each student on almost a daily basis, which would be almost impossible to falsify. In practice, the (online) characters of students become very well-known to their instructors, other students and to the module monitors.

It remains possible, of course, for students to plagiarise their programming assignments (just as is the case in 'on-ground' teaching). However, the fact that all communications take place online, and are recorded and preserved for as long as required (i.e. at least until after student graduation) again provides some protection. In particular, it becomes relatively easy to apply programs (e.g., [7] that perform comparisons of work submitted in the Virtual Classroom, or use services that perform checks against plagiarism throughout the Web.

5. PRACTICAL EXPERIENCE

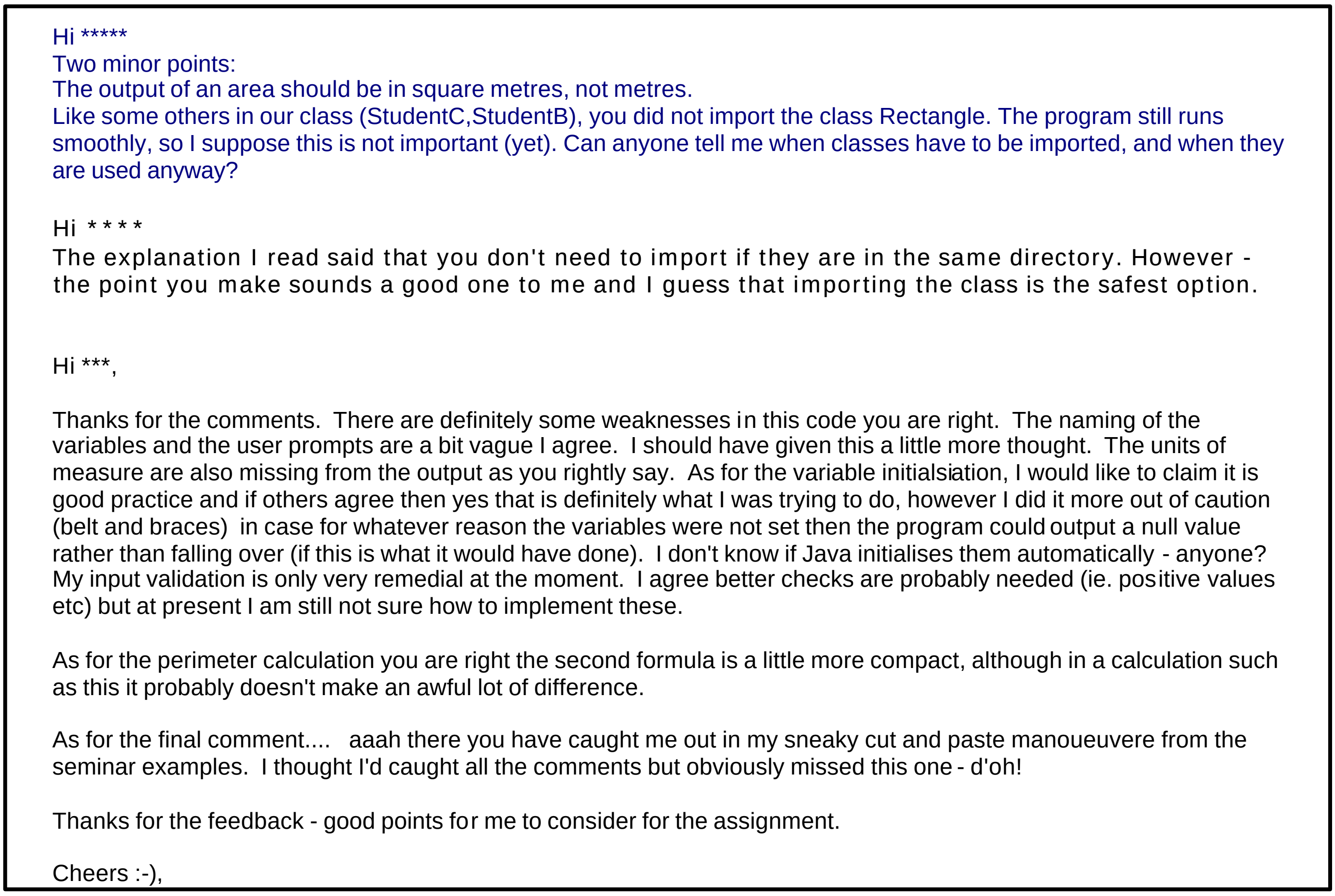
Although the technology plays a key role in the delivery of distance education, the focus in the Java module is very much on learning outcomes. The key to effective distance education is focusing on the needs of the learners, the requirements of the content, and the constraints faced by the teacher.

One of the challenges when instruction is delivered at a distance is the fact that students are separated from others sharing their backgrounds and interests, they often have few opportunities to interact with teachers outside class and they must rely on technical linkages to bridge the gap separating class participants. The Java class is made up of students from a variety of backgrounds and with differing levels of programming experience. Every effort is made to keep the instructional content simple enough for those just beginning to program and challenging enough for those with more experience. It is a case of getting to know the characteristics of the class and then adapting teaching style as a consequence. Targeting material is always difficult but the nature of the online environment means that the instructor can fill in any gaps by redirecting the discussions or providing pointers to supplementary material, answering questions personally etc.

There are often a few heated discussions as time progresses and the instructor has to mediate between group members but this is as a result of normal group development and everyone establishing themselves. The group eventually develops a sense of community and each student finds their place. Sometimes however, the discussions dry up or go off at a tangent and therefore another challenge is to keep the interaction momentum going and increase collaborative learning. Programming is essentially a practical activity and the burden of emphasis in the Java module is placed on the practical element. This means that students have to provide samples of the programs that they have written in response to the discussion questions. Lack of experience with Java coding and shyness means that some people are often reluctant to comment on other people's work or fear comment on their own. The distance teacher has few cues and in our case none that are visual. Those cues that do exist are filtered through the technological media – it is difficult to carry on a stimulating teacher-class discussion when spontaneity is altered by technical requirements and distance. Separation by distance affects the general rapport of the class. The instructor has to keep everyone talking and participating in the process by introducing new questions, by pointing out general trends in the style of submissions and by restating the ground rules for interaction in order to encourage students who lack confidence in their programming abilities. Students need to get comfortable communicating – trust that it will be professional and polite and that their communication is valued and required.

In an eight week course there is no time to waste. One of the most difficult challenges to overcome in any class is the difference between the learning rates of students. The diversity of students attending the Java course, their learning styles and varying ability to cope with the interaction method becomes really evident to the instructor as time goes on. Having taught in a traditional classroom setting previously, some instructors find that it is somewhat easier to spot potential problems in the VC. One of the most positive outcomes of the course is that students feel increasingly able to express their concerns openly as the weeks progress because a sense of community has been fostered. The result of this trust and openness is usually an increase in interaction and enthusiasm. The instructor is able to get to know students individually, their pattern of interaction, their main concerns and their learning achievements as the course progresses. This gives greater insight into student expectations and allows the instructor to respond accordingly with encouragement and advice. Programming in Java proves an ideal subject matter to teach online as students have programming tools and literature readily available at a keystroke. They find it easy to discuss material and introduce new material themselves via their research on the Internet etc. and this makes the experience richer than it would perhaps have been in the traditional classroom setting.

As group development and interaction increase, students eventually reach a stage of knowledge construction in which they are highly productive and collaborative learning begins – the ability to share differing views, personal experiences and abstract ideas develops – here the learning process is truly an on-line experience as members are learning from each other, more than they are socialising and conflicting. This is where the core learning takes place and there is a flattening of communication between instructor and students – the learning field becomes more equal where both student and instructor facilitate learning. Below are sample extracts from a discussion thread regarding solutions to a problem on calculating the area and perimeter of a rectangle, as an example of how students begin to facilitate learning themselves:



Hi *****
Two minor points:
The output of an area should be in square metres, not metres.
Like some others in our class (StudentC,StudentB), you did not import the class Rectangle. The program still runs smoothly, so I suppose this is not important (yet). Can anyone tell me when classes have to be imported, and when they are used anyway?

Hi * * * *
The explanation I read said that you don't need to import if they are in the same directory. However - the point you make sounds a good one to me and I guess that importing the class is the safest option.

Hi ***,

Thanks for the comments. There are definitely some weaknesses in this code you are right. The naming of the variables and the user prompts are a bit vague I agree. I should have given this a little more thought. The units of measure are also missing from the output as you rightly say. As for the variable initialsiation, I would like to claim it is good practice and if others agree then yes that is definitely what I was trying to do, however I did it more out of caution (belt and braces) in case for whatever reason the variables were not set then the program could output a null value rather than falling over (if this is what it would have done). I don't know if Java initialises them automatically - anyone? My input validation is only very remedial at the moment. I agree better checks are probably needed (ie. positive values etc) but at present I am still not sure how to implement these.

As for the perimeter calculation you are right the second formula is a little more compact, although in a calculation such as this it probably doesn't make an awful lot of difference.

As for the final comment.... aaah there you have caught me out in my sneaky cut and paste manoueuvere from the seminar examples. I thought I'd caught all the comments but obviously missed this one - d'oh!

Thanks for the feedback - good points for me to consider for the assignment.

Cheers :-),

Figure1: Example of student facilitated learning in the VC.

At this stage of learning the role of the facilitator becomes more specific – they need to weave common ideas and thoughts together as they apply to the objectives, narrowing in on points that hit home the concepts – this is a method that allows further exploration and development of issues. Any form of leading the group to a higher understanding is appropriate here. The instructor must be actively engaged with the streams of thought that develop in the VC. The discussions in the VC are just that – streams of thought. Discussion is the best opportunity or tool that the instructor has in terms of testing, measuring, challenging and enhancing the learning that is taking place and the most work is accomplished at this stage.

Participants become responsible for their own learning and need little support – critical thinking is dominant, along with the ability to challenge what is happening – they become more demanding and resistant to changes and experienced participants become more helpful as guides to those less experienced, students begin to question their own thinking processes and this includes both topic areas and also the teacher's and student's responses. Higher-level skills begin to develop and the facilitator supports this learning and responds to all questions and concerns as they occur.

Plans for further adaptation for the Java module include the introduction of the broader concepts of Object-Oriented programming in comparison to other paradigms as students felt the course material did not cover this in enough detail. Students also felt that the course text was not comprehensive enough and those students with more experience in programming sought greater challenges and the incorporation of material on Java Beans etc. The instructors find it very useful to receive this feedback from students so the course can be further developed and improved.

6. CONCLUSIONS

We have here reported on our initial experiences from the delivery of a Java programming module to MSc students being taught entirely through the medium of the Internet. Although the programme is as yet at an early stage – to date, no students have graduated – reactions so far have been very positive. Contrary to many preconceptions, online learning, in the mode we have described, is very far from being an impersonal and alienating experience. On the contrary, we believe that both staff and students find it to be a stimulating and challenging mode of teaching and learning, which has more in common with small-group seminar-based learning than it has with conventional lecture-based teaching and other methods of distance learning.

A key element is the role played by discussion in the Virtual Classroom. In a course such as this, many of the students bring to the VC a wealth of personal experience from their earlier studies and their professional life, often including knowledge outside the scope of their instructors. At the simplest level, this often means that simple queries about Java are answered by other students, lessening the load on the instructor. Beyond this, the mediated classroom discussion provides a means in which students can share their broader knowledge with their colleagues, enriching the learning experience for everyone.

REFERENCES

1. E Gruengard, Y M Kalman and P H Leng, "University education via the Internet: a new paradigm for public-private partnership", in 'Innovation through Electronic Commerce' (Proc IeC '2000 Conference), ed. L Macauley, CeeC, Manchester, 46-53
2. Hiltz, S.R. and Wellman, B. (1997) "Asynchronous Learning Networks as a Virtual Classroom", *Comm ACM* 40(9), pp 44-49
3. Persico, D. and Manca, S. (2000). "Use of FirstClass as a Collaborative Learning Environment". *Innovations in Education and Training International,* 37,1, pp 34-41
4. Wilson, B.G. (1996) "Constructivist Learning Environments: Case Studies in Instructional Design". Educational Technology Publications.
5. Klemm, W.R. and Snell, J.R. (1996) "Enriching Computer-Mediated Group Learning by Coupling Constructivism with Collaborative Learning", *Journal of Instructional Science and Technology* 1(2)
6. Hubbard, J.R. (1999). Schaum's Outlines: Programming with Java. McGraw-Hill. ISBN 0-07-134210-9
7. Joy, M. and Luck, M. "Plagiarism in Programming Assignments". IEEE Trans Education, 42(2) (1999) pp. 129-133