# Using tutor and demonstrators to support an in-class open book assessment

**Christos Petichakis, Stuart McGugan**
Educational Development Division, Centre for Lifelong Learning,
University of Liverpool
c.petichakis@liverpool.ac.uk, s.mcgugan@liverpool.ac.uk
http://www.liv.ac.uk/eddev

## OVERVIEW

This case study relates to the conference theme of *Assessment and the first year experience* and reports on a Computing module assessment where the learning of students and the practice of programming skills in software development is facilitated through a series of 'in-class' open book assignments. Introduced in the academic year 2005-2006, the assessment was designed in response to growing staff concerns over plagiarised coursework as well as a perceived weakness in programming skills development. A particular feature of this assessment is the use of a tutor supported by trained demonstrators who engage productively with the students as the 'in-class' assessment takes place. This provides the students with prompt feedback and the tutor with an insight as to the level of competence of the whole class.

## INFORMATION ABOUT THE CLASS, MODULE OR PROGRAMME

*Introduction to Programming in Java* is a 15 credit module and takes place in the first year, first semester of study of the Bachelor of Science in the Computer and Multimedia Systems programme at the University of Liverpool. It aims to develop the students' understanding of the Java programming language and to acquire fundamental software development skills covering program design, coding and testing. On average, about 40 undergraduates take this module every year over a 12 week semester. The module is continuously assessed using various assessment activities throughout the semester.

## DESCRIPTION OF THE CASE

The module is presented to the students through a series of one hour lectures followed by three hours of laboratory practice. During the lecture, the tutor discusses software theories relevant to the Java language and the use of narrative analysis of examples allows the students to understand the development of a software application. In the laboratory, the students have the opportunity to work on problems that are relevant to the material presented in the lecture. During the laboratory time, demonstrators are available to support and guide the students. The demonstrators are postgraduate (usually PhD level) students and are familiar with Java programming.

For this module, it was decided to alter the core of the existing assessment due to the reported high number of plagiarism cases. The majority of the marks of the original assessment could be acquired through 3 home assignments (Appendix 1, Table 1). The new assessment demoted the significance of the home assignment and introduced 'in-class' assignments (Appendix 1, Table 2) with the aim to assess students' programming skills in a *monitored* and *supportive* environment. The 'in-class' assignments took place, instead of the scheduled weekly laboratory sessions, in weeks 4, 8 & 11 and lasted 3 hours. In weeks 3, 6 & 9 the students had to complete 3 brief home assignments and in week 7 they participated in a multiple choice class test that assessed the theoretical aspect of the module. In addition, at the end of each laboratory session, the students had to have the practice problems they completed on the day signed by a demonstrator. In Appendix 1, Figure 1 shows a graphical representation of the key milestones, the students' activities and the marks distribution over the semester.

## RATIONALE IN TERMS OF EDUCATIONAL IDEAS

The assessment method, the learning of students and the practice of programming skills in software development modules have been major issues of concern in Universities for a number of years. Hilton et al. (1997) presents evidence that first year students do not receive the necessary skills and the appropriate computing education. Although they manage to progress to the next year of study, students often find it extremely difficult to cope with level 2 computing modules. The problem receives an extra dimension as industry employers have repeatedly expressed their discontent and a growing frustration. They are unhappy with the quality of computing graduates who aim to become professionals and compete in a demanding environment controlled by tight deadlines and diverse needs of stakeholders (Calahan & Pedigo, 2002).

The assessment of the Java module utilises a Continuous Assessment (CA) model. For a computing module, the advantage of CA is that it assesses the software design, development and testing skills of the student progressively. Feedback on the development of these skills can then be provided by the tutor who is in an excellent position to identify gaps in learning and suggest improvements for the remainder of the semester (Trotter, 2006). Additionally, in order to complete the assignment, the students have the opportunity to research the problem, engage productively with relevant work and operate in an environment fostered by support, including from their peers.

In reality however, widespread plagiarism was observed. This was perhaps due to the fact that 70% of the assessed work (Appendix 1, Table 1) was undertaken during the students' own time. As a result the learning of a large number of students was compromised. For that reason, the necessity to modify the assessment of the module became apparent.

The decision to modify the existing assessment was based on the rationale that,

- Students should operate in an environment that replicates a professional setting
- Plagiarism, due to high volume of home assignment, was compromising the learning of the students
- A prompt feedback mechanism was needed to underline students' learning

The proposed new assessment can be seen in Appendix 1, Table 2. Its key characteristic is the use of 'in-class' assignments. In addition, a set of guidelines have been created which are then presented and discussed with the students during the lecture to ensure that everybody is aware of the new format. These guidelines are as follows:

- All 3 'in-class' assignments are open book - any written material and the internet can be used
- Students are not allowed to engage with their peers during the 'in-class' assessment
- The assessed exercises are different for each student - a pool of themed questions was created, they were then added to the assessment script in a random order and given to each student

More specifically, with guideline 1 the students are encouraged to bring any material they think is necessary and use it freely in the class. According to Theophilides & Dionysiou (1996) an open book examination 'reduces examination tension and stress and leads to lasting learning outcomes'. The students feel more comfortable as they have their choice of support material. In addition to this, in terms of their learning, they have to identify the resources they need and modify them accordingly to fit in with the programming code they develop. As a result, the students learn to research and critique whether the information they have is suitable and how they can apply it. Such a learning environment is comparable to a professional setting where staff adapt existing methodologies to fit in with their project. Moreover, Feldhusen (1961) noted over 40 years ago that open book examination reduces the unnecessary rote of memorising. This argument is particularly relevant in programming modules where the material and resources available are evolving rapidly and existing practices soon become obsolete. It is therefore imperative for students to have the skill to learn how to apply the existing practices and not to memorise facts that are valid at the moment, but which may become out of date very soon.

However open book exams are not without their critics. For example Feller (1994) argues that 'if students know in advance that their tests will be open book, they will not study for the tests'. Although this can pass through the students' mind, the input of the tutor in this case addresses this issue. Prior to the assessment, the tutor must make explicit to the students the purpose of the open book assessment and the environment under which the students will have to operate in order to succeed. Guidelines 2 and 3 set the rules when the students are in the computer laboratory and engage with the assessment. The aim of these rules is to ensure that the assessment is fair to all the students and students will receive marks based on their own merit and their individual preparation.

At this point, it can be argued that although the plagiarism issue may have been addressed, the students may still engage with the assessment at a surface learning level. According to Smith (2002) surface learning focuses on a minimal grasp of ideas. From this it can be argued that the students will do their best to complete the questions without actively reviewing and evaluating the information they plan to use. Markedly, this scenario may take place because at this stage the students are working in isolation. It was therefore considered essential to involve the tutor in the open book assessment. Hence, as a 4th Guideline the tutor and trained demonstrators engage productively with the students as the 'in-class' assessment takes place. It is crucial to emphasise that for this model to work effectively, the demonstrators, who themselves are postgraduate students, are already knowledgeable with the assessment material and the tutor has explained clearly their exact role during the assessment. Moreover the tutor needs to ensure that the support the students receive is consistent and reliable. If a problem arises between a demonstrator and a student, the demonstrator must inform the tutor immediately. It is then the tutor's responsibility to ensure that the student is given the necessary attention with the aim to keep the disruption to an absolute minimum and allow the student to proceed with the assessment. Tutor and demonstrators make themselves available to every student. Their function as a 'support team' is to offer guidance and tips. The tutor and the demonstrators are also readily available to support, encourage and motivate the students and offer prompt feedback (explicitly on the students' approach to problem solving). This accords with Chickering and Gamson (1991) contention that the sooner feedback is offered to the student the better it is for the student's learning. When implementing this practice it should be noted that the tutor and demonstrators should not propose a solution but rather seek to promote the students' independent learning. If necessary they should also seek to engage the student in a productive dialogue around the problem (Nicol and Macfarlane-Dick, 2006) which in turn should initiate a deep learning process on behalf of the student (Biggs, 1989). In such cases, the student is acting as the pioneer with the tutor and demonstrators acting as advisors. According to Jarvis et al. (2003) this brief interaction between the tutor and the student will reveal the level of understanding of the student. Moreover the fact that the tutor is available to every student can help the tutor to gain an insight into the quality and level of learning taking place.

## EVALUATION

In terms of evaluating the assessment, it is interesting to compare the average mark of the module before and after the assessment changed. The previous mark average was 53%, the new mark average after the introduction of the 'in-class' assessment was 55%. Although the overall average was slightly improved it can now be argued that the students *own* this mark. In addition it was observed that the students went through an assessment model that required dedication, good preparation and the ability to blend teaching and learning in a supportive environment. They had the opportunity to engage productively with the tutor and the demonstrators during the 'in-class' assignments in a way that promoted consistency and reliability. The tutor/demonstrators were aware of the assessment method and were available to each student as an interactive support mechanism. Moreover, the validity of the assessment was evident as it captured the learning outcomes of the module. In particular, the students were given the opportunity to communicate in order to receive guidance, deal with the large amount of information available to them appropriately and work on problems that required sound technical skills. Additionally, the remaining assessment methods used in the module, allowed the students to develop new or existing skills, for example adhere to precise instructions and time management during the multiple choice assessment, or team work and project management during the home assignments and the laboratory activities.

## REFERENCES

Biggs, J., B. (1989) Approaches to the enhancement of tertiary teaching. *Higher Education Research and Development* 8, 7-25.

Callahan, D., Pedigo, B. (2002) Educating Experienced IT Professionals by Addressing Industry's Needs. *IEEE Software* 19 (5), 57-62.

Chickering, A., W., Gamson, Z.F. (1991) Applying the Seven Principles to Good Practice in Undergraduate Education. San Francisco: Jossey-Bass.

Feldhusen, J., F. (1961) An evaluation of college students' reactions to open book examinations. *Educational and Psychological Measurement* 21, 637-646.

Feler, M. (1994) Open-Book Testing and Education for the Future. *Studies in Educational Evaluation* 20, 235-238.

Hilton, M., Samy, N., Grant, D., Greening, T. (1997) Software Engineering Education – Is it Meeting Industry Needs? Can Industry Needs Be Met? *Software Engineering Conference Proceedings,* Motorola Australia Adelaide.

Jarvis, P., Holford, J., Green, C. (2003) *The Theory and Practice of Learning*. 2nd Edition, London: Routledge Falmer.

Nicol, D., J., Macfarlane-Dick, D. (2006) Formative Assessment and Self-Regulated learning: A Model and Seven Principles of Good Feedback Practice. *Studies in Higher Education* 31(2), 199-218.

Smith, J. (2002) Learning Styles: Fashion Fad or Lever for Change? The Application of Learning Style Theory to Inclusive Curriculum Delivery. *Innovations in Education and Teaching International* 39 (1), 63-70.

Theophillides, C., Dionysiou, O. (1996) The Major Functions of the Open-Book Examination at the University Level: A Factor Analytic Study. *Studies in Educational Evaluation* 22 (2), 157-170.

Trotter, E. (2006) Student Perceptions of Continuous Summative Assessment. *Assessment and Evaluation in Higher Education* 31 (5), 505-521.

## APPENDIX 1

The original assessment of the Java module

**Table 1. Module assessment**

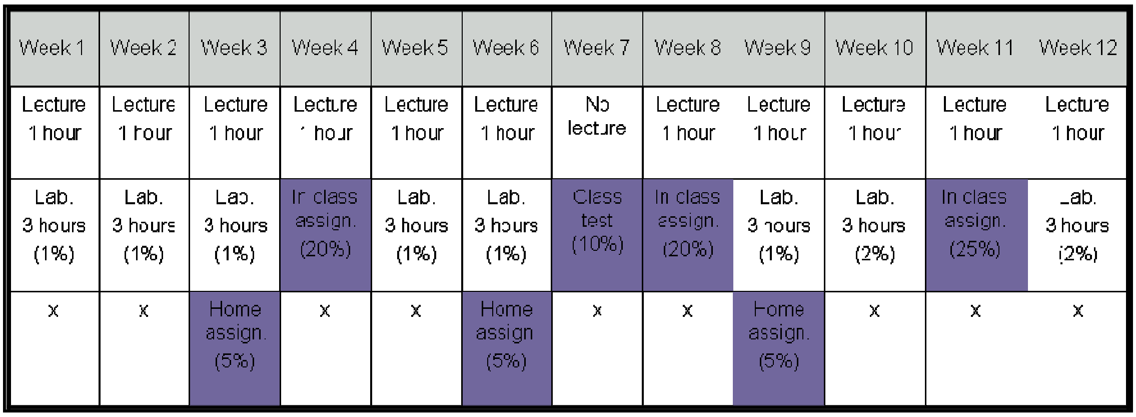
| Assessment element | % of total mark |
|---|---|
| 3 homework assignments | 70 |
| Multiple choice class test | 20 |
| Laboratory practice exercises | 10 |

The proposed new assessment of the Java module

**Table 2. New Module assessment**

| Assessment element | % of total mark |
|---|---|
| 3 ‘in class’ assignments | 65 |
| 3 homework assignments | 15 |
| Multiple choice class test | 10 |
| Laboratory practice exercises | 10 |

**Figure 1. Graphical representation of key milestones and students' activities through the semester. (In brackets the assessment weight for each activity)**



| Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Lecture 1 hour | Lecture 1 hour | Lecture 1 hour | Lecture 1 hour | Lecture 1 hour | Lecture 1 hour | No lecture | Lecture 1 hour | Lecture 1 hour | Lecture 1 hour | Lecture 1 hour | Lecture 1 hour |
| Lab. 3 hours (1%) | Lab. 3 hours (1%) | Lab. 3 hours (1%) | In class assign. (20%) | Lab. 3 hours (1%) | Lab. 3 hours (1%) | Class test (10%) | In class assign. (20%) | Lab. 3 hours (1%) | Lab. 3 hours (2%) | In class assign. (25%) | Lab. 3 hours (2%) |
| x | x | Home assign. (5%) | x | x | Home assign. (5%) | x | x | Home assign. (5%) | x | x | x |

Re-Engineering Assessment Practices in Scottish Higher Education (REAP) is funded by the Scottish Funding Council under its e-Learning Transformation initiative. Further information about REAP can be found at http://www.reap.ac.uk