Methods for Evaluating the Appropriateness and Effectiveness of Summative Assessment via Multiple-choice Examinations for Technology-Focused Disciplines

Raymond Lister
University of Technology, Sydney, Australia

We describe an overall approach to setting objectives which is based on Bloom’s taxonomy, then concentrate on the assessment of students at the Knowledge and Comprehension levels of that taxonomy, for which students may receive the minimum passing grade. Multiple-choice questions are a suitable way of assessing that level of competence. We analyze student performance on such an exam, using well established analysis techniques from the literature on multiple choice questions. The examination was found to have largely met its objectives, but some problems are noted.

Introduction

... the lecturer must guide this collection of individuals through territory the students are unfamiliar with, towards a common meeting point, but without knowing where they are starting from, how much baggage they are carrying, and what kind of vehicle they are using. This is insanity. (Laurillard 1993, p. 3)

When assessing their students, most academics tread a thin line between establishing that the students have met the official subject objectives, while ensuring that the resultant failure rate is politically acceptable. Academics who teach early in degree programs are particularly exposed to scrutiny, as their colleagues who teach “downstream” can be relied upon to make their displeasure known if students are subsequently found to be lacking fundamental skills: an engineer teaching downstream may complain about the mathematics skills of students, while a “downstream” lecturer in construction & design might complain about the students’ ability to draw a correct plan.

Within the discipline of information technology, the common complaint among downstream teachers is focussed on the programming skills of students. The complaint is not simply that many students lack the more advanced skills, but that the weaker performing students even lack elementary skills. However, if the poor performing first year students are failed in first year, the failure rate can be unacceptably high. Across the world, it is well recognized that novice programmers struggle with elementary programming skills. Results from a recent project by McCraken et al. (2001) are compelling, because of the number of authors from differing educational institutions and cultures. Four of the authors, from three countries, tested their students on a common set of programming tasks. The students performed much more poorly than the authors had expected. The students did not simply fail to complete the set task; most students did not even get close to solving the task.
But if we focus our teaching on the weaker performing students, the better students are not challenged. While most students studied in the McCracken project performed poorly, some students performed quite well. The considerable diversity of abilities of incoming students is a challenge that is not peculiar to teachers of information technology. However, the writing of subject objectives and assessment design rarely distinguishes between expected outcomes for students who barely pass and students who excel.

In the next section, I will outline an approach to objective writing and assessment, based on Bloom’s taxonomy. This approach separates explicitly the objectives and the assessment activities of low and high achieving students. The remainder of the paper will the on focus on the assessment of the low achieving (but passing) students, who are assessed primarily via multiple choice examinations.

** Appropriateness: Multiple Choice Examinations and Bloom's Taxonomy**

Bloom's taxonomy contains six levels, with the organising principle that competence at a higher level of the taxonomy implies a reasonable degree of competence at the lower levels (Bloom, 1956). Furthermore, successive levels of the taxonomy are paired, to form three groups, with qualitatively different assessment standards expected between the different groups. The remainder of this section briefly reviews those three groups, from lowest to highest. A detailed description of this Bloom-based assessment approach, aimed at members of the information technology community, has been published elsewhere (Lister, 2005; Lister and Leaney, 2003a, 2003b, 2003c). The following brief review does not assume deep knowledge of information technology.

**Knowledge and Comprehension.** Informal descriptions of the taxonomy frequently confuse the two lowest levels of the taxonomy. At the lowest “knowledge” level, a student can regurgitate a fact when prompted for it, without necessarily understanding the significance of the fact, which can simply be achieved via rote learning. Higher education does not advocate the encouragement of rote learning (at least not rote learning for memorization; a topic which I will discuss later). The next level of the taxonomy is the “comprehension” level. It is a higher level because a student competent at the comprehension level understands the significance of a fact. A student manifests that understanding by supplying knowledge when prompted for it in a way that is different from how the material was first taught. For example, in the case of computer programming, a student would be required to demonstrate their mastery of a concept by correctly identifying its use in a piece of computer code not previously seen by the student. We have found that multiple choice questions are entirely suitable to assessing students for this outcome. At the author’s university, there are four passing grades, the lowest of which is the "Pass". Students who demonstrate competence at the knowledge and comprehension levels, via such an examination, are given this minimal passing grade.

**Application and Analysis.** At these intermediate levels of the taxonomy, students are expected to be able create and analyze artefacts, but within a well defined context. For example, in the case of computer programming, a student might be required to take an existing program, identify salient features, and complete a small and well defined modification to that program. Students who demonstrate competence at this level are awarded one of the middle two passing grades, the “Credit” or the “Distinction”.

**Synthesis and Evaluation.** At this highest level of the taxonomy, students are expected to show considerable skill in setting and achieving their own goals, with minimal assistance from the teacher, and also show critical skills in evaluating artefacts. For example, in the case of the computer programming subject described in this paper, students operating at this level...
are expected to choose their own programming project that demonstrates the key concepts of the subject, and write the entire program. Furthermore, these students must also peer review each other’s projects. Students who demonstrate competence at this level are awarded the highest grade, the “High Distinction”.

**Design Assumptions and Objectives for the “Pass” and the Multiple Choice Examination**

The multiple-choice examination described here is one of a number of assessable tasks undertaken by the students in this subject. Its design was based on the following assessment principles.

*The Multiple Choice Examination is a Test of Mastery, Not Development*

The aim of the multiple-choice examination is to identify those students who have mastered the essential concepts and skills required of all passing students. The aim is not to test their level of development. That is, the aim is not give students the opportunity to perform to the very limit of their ability (Linn and Gronlund, 1995). The other assessment components, aimed at levels of Bloom’s taxonomy higher than Knowledge and Comprehension are intended to challenge the student.

*The Minimum Acceptable Passing Score in the Multiple Choice Examination is 70%, Not 50%*

The traditional but arbitrary pass mark of 50% often allows students to pass who have a weak grasp of the subject material, especially since the examination is multiple choice.

*Balance Formal Criterion-Referencing and Real-Politick Norm-Referencing*

At the author’s university, it is policy that assessment is criterion-referenced. That is, a grade is awarded to a student if the student meets the criteria for that grade, irrespective of how many other students have been awarded the same grade.

In practice, however, every experienced teacher knows that their faculty/department/school has expectations of what the grade distribution will be, especially the failure rate. In the author’s faculty, a failure rate above 30% is considered too high. In fact, the actual examination presented later in this paper had a failure rate closer to 50%. To achieve the 30% figure, the passing criterion would need to be lowered to 50%, in contradiction of the preceding design assumption (the pass mark was not lowered). The analysis of the examination presented later in this paper will compare the detailed performance of students who scored around 70% versus students who scored around 50%.

*Assess Minimal Competence in Both of Two Loosely Coupled Halves*

While the subject is concerned with programming, it combines two different styles of programming. This is reflected directly in the examination, with the first half of the questions on “object-oriented” programming, the second half on “procedural programming”.

Combining loosely coupled elements into a single subject is common in technical degree programs, particularly in the early “core” subjects. Downstream lecturers tend to assume that students are competent in all the loosely coupled elements, and that is the ideal outcome from the upstream subject, but the danger is that students may pass the examination as a whole by doing particularly well on a subset of the elements. A minimum pass mark could be specified for each loosely coupled element, but that greatly increases the risk of a blow out in the
failure rate, and also leads to supplementary assessment and/or appeals process that is debilitating for the teacher.

Assess Minimal Competence in the Designated Algorithms

Within the procedural component of the subject, an objective is to teach certain well known, basic algorithms, known as the “sorting and searching” algorithms. This is reflected directly in the examination. Of the 13 procedural questions, the final 8 questions directly test students recognition of these algorithms. Furthermore, these 8 questions are drawn from a pool of questions; to which students have access during semester, and which are used in tutorials as part of the learning process. Prior to the examination, students were told that eight questions in the pool would appear in the examination. We refer to these final 8 questions as the “seen” questions.

Student Preparedness for the Examination: Summative vs. Formative Assessment

This paper focuses on the activities which contribute directly to the grading of the students (i.e. the summative assessment). This paper does not describe the extensive activities which students undertake as part of the learning process, but for which marks are not assigned (i.e. the formative assessment; Bloom, Hastings & Madaus, 1971). For example, each week students were given weekly lab exercises, and weekly tutorial exercises, but these did not contribute to the grading.

Several weeks prior to sitting the examination, students were given three complete multiple choice examinations from earlier semesters. All three examinations are similar in style to the examination paper analyzed in this paper. Furthermore, throughout semester many tutorial questions were taken from these past examination papers. Therefore, students were well prepared for this style of examination.

The Examination Should Not Reward Rote Learning

Given the conditions specified in the above two subsections—the need to test some very specific material in the procedural component of the subject, and the extensive access students have to questions on that material—it is easy to conceive the danger that students will rote learn the answers to those final eight “seen” questions. This danger, however, may be easily overstated. Most experienced programmers fail to appreciate the high cognitive load needed by novice programmers to read and memorize code. When reading code with a view towards “remembering” that code, experienced programmers tend not to remember low level detail minutia. Instead, they “memorize” the purpose of the code at a more abstract level. When reproducing that code, experienced programmers reconstruct the code from that abstraction. In contrast, novice programmers struggle to see the high level purpose of code, only the concrete code before them. This is particularly so for students finding it difficult to meet the subject’s minimums requirements. The author of this paper led an international project (Lister et al., 2004) which asked students to “think out loud” (Bloom and Broder, 1950) as they read program code. The pieces of code used came directly from past examinations used and written by this author. In terms of the SOLO taxonomy (Biggs & Collis, 1982), students tended to articulate the code at structural and multistructural levels. In contrast, expert programmers were found to typically articulate at the relational level.

The difference in novice and expert memorization styles is not peculiar to computer programming. Psychologists have described the use of semantic “chunking” by chess experts when remembering board positions (Simon, & Barenfeld, 1969; Simon & Gilmartin, 1973; Chase & Simon, 1973). Within educational psychology, a distinction is made between meaningful reception learning (Ausubel, 1963) and rote learning (Lefrancois, 1999, pp. 213-219). Within the teaching and learning literature, there has been considerable interest in these
differences, sparked by the observation that many highly successful students from a Confucian tradition appear superficially to employ rote learning, but on closer analysis they may employ memorization as a preliminary process in moving toward genuine understanding (Au & Entwistle, 1999; Marton, Dall’Alba & Tse, 1993; On, 1996).

The first five procedural questions in the examination (i.e. questions 14–18) are completely “unseen” by the students prior to the examination. Those five questions, while also testing other objectives, are also partly intended to check on whether students are successfully rote-learning the final eight questions, or have developed a deeper understanding of the material underlying the “seen” questions.

**Evaluation of the Multiple Choice Examination**

This section reviews analyses the multiple-choice examination to determine whether it meets the above design assumptions and objectives. A discipline-specific review of these types of examination questions has been done elsewhere (Lister, 2000, 2001, 2005). The following review is not discipline specific.

Figure 1 shows three graphs. Each graph summarizes the performance of the student groups on each of the 26 multiple-choice questions. The left graph shows the performance of the class as a whole on each question. There is greater variation in class performance on the first half of the examination, the 13 object-oriented questions, than the remaining procedural questions, and in general, the class did better on the procedural questions than the object-oriented questions. Within the procedural questions, the class did a little better on the “unseen” questions than the “seen” questions.

The middle graph of Figure 1 separates the students into four quartiles, based on their scores on the entire examination, which is a well-established way of analyzing multiple choice questions (Ebel & Frisbie, 1986; Linn & Gronlund, 1995). The right graph summarizes the performance of two subgroups of students of particular interest: the 42 “bottom passing” students (a mark of 18-19), represented by squares, and “fifty-fifty” students (mark of 12-14), represented by diamonds, who answered each question correctly.

The middle graph of Figure 1 separates the students into four quartiles, based on their scores on the entire examination, which is a well-established way of analyzing multiple choice questions (Ebel & Frisbie, 1986; Linn & Gronlund, 1995). The right graph summarizes the performance of two subgroups of students of particular interest: the 42 “bottom passing” students (a mark of 18-19, just at or over 70%), and 38 “fifty-fifty” students (mark of 12-14, a mark of around 50%). Within each of the three graphs, there are two vertical lines. Questions to the left of the first vertical line are the 13 object-oriented questions. Questions between the vertical lines are the procedural “unseen” questions. Questions to the right of the second line are the procedural “seen” questions.

**Validity and Problematic Questions**

A question-by-question examination of the left and middle graphs of Figure 1 leads to the identification of some problematic questions, discussed below.

**Unfair (“trick”) questions.** Only 22% of the whole class answered Question 1 correctly. Furthermore, only 40% of the top quartile answered correctly. This was the only question in
the examination where a minority of top quartile students answered correctly. We conclude that this was an unfair question, but it is (by these criteria) the only unfair question in the examination.

**Easy questions.** Question 4 proved to be the easiest of the thirteen object-oriented questions, with 84% of the whole class and even a majority of the bottom quartile students answering correctly. However, the bottom passing students performed much worse on questions 5, 6 and 7, which are conceptually similar to question 4, which is evidence that students of lower ability answered question 4 by an approach that did not require genuine understanding of the underlying concepts.

**Guessed Questions.** In questions 8 and 13, performance difference narrows between the quartiles. The performance of the upper quartile is relatively low, while the bottom quartile performs comparatively well, suggesting that all students chose an answer not from knowledge, but by guesswork.

**Point-By Point Evaluation of Satisfaction of the Design Objectives**

This section compares the design assumptions outlined above with results reported in Figure 1.

**The Multiple Choice Examination is a Test of Mastery, not Development.** The top quartile consistently performed above 90% on the procedural questions, and the second quartile consistently performed above 80% on the same questions. While not committed to the specific figures of 90% and 80%, the author believes that the high performance of these two quartiles is justification that the array questions are a reasonable test of mastery. Both of the upper two quartiles performed less consistently on the object oriented questions, suggesting that this portion of the examination is a less reasonable test of mastery.

**The Minimum Acceptable Passing Score in the Multiple Choice Examination is 70%, not 50%.** The fifty-fifty students performed particularly poorly on the “seen” procedural questions, especially when compared to their performance on the unseen procedural questions, suggesting that they lack an understanding of underlying concepts.

**Assess Minimal Competence in Both Two Loosely Coupled Halves.** Even the top quartile students performed poorly on many of the object-oriented questions, suggesting that the examination was not successful in testing mastery of this half of the course.

**Assess Minimal Competence in the Designated Algorithms.** Bottom passing students performed reasonably well on the “seen” questions, compared to their performance on the “unseen” questions.

**The Examination Should Not Reward Rote Learning.** The low scoring students performed relatively poorly on the “seen” questions compared with the “unseen” questions.

**Streaming**

A very serious issue that results from the analysis of this examination is whether students who only achieve the lowest passing grade are subsequently disadvantaged. That is, having achieved only a minimum pass in the first semester programming subject, are those students condemned to nothing higher than a minimal pass in later semesters? This is not the intention of this assessment scheme. On the contrary, a design assumption of the scheme is that if students are solidly grounded in the fundamentals of programming, then they are equipped to...
possibly excel in later semesters. Data confirms that this is the case. In a particular semester, we looked at the students who achieved one of the passing grades in the 3rd semester programming subject, and who had recorded only a minimal passing grade when they completed the first semester programming subject two semesters earlier. Of those 50 students, only 16 recorded a minimal passing grade again. The remainder recorded a higher passing grade, including 3 students who recorded the highest possible grade ("High Distinction").

Student Satisfaction

At the authors' university, the students enrolled in all subjects are surveyed every semester. That survey contains the statement, "I found the assessment fair and reasonable", to which students respond on a five point Likert scale. In one semester (not the semester from which the above examination statistics were derived, but a semester with a very similar structure and examination), 64% of students agreed with that statement, 30% responded neutrally, and only 7% disagreed. A 64% agreement is a high rate of agreement for an information technology subject at the author's university.

It is interesting that almost one in three students responded neutrally on this question. Perhaps this is due to students being surveyed in week 10 of semester, well before much of the summative assessment had been marked and returned. Even so, what does a student mean when they respond neutrally to this question? Are they truly neutral, or do they mean that, based upon prior experience, the assessment in the subject is what they have come to expect? Since the survey instrument is a standard university-wide survey, and given the requirement that students answer the survey anonymously, it is difficult for the author to explore these interesting survey results further. For example, a breakdown of the student response by grade awarded might be illuminating, but as the survey is anonymous, no such breakdown is available. Given the importance of these surveys in universities, across all disciplines, it is remarkable that there has not been more research into what students are thinking when they answer these surveys: what does "fair and reasonable" mean for students, and does "fair and reasonable" mean something different for academics?

Conclusion

Most academics lead double lives. Their research life is communal, they read the relevant literature, attend seminars and conferences, and work within well-defined theoretical or empirical frameworks. Their teaching life, however, remains private, and not so well informed; they are guided only by their past experience and their intuition. This private teaching life has copiously documented disadvantages for students. What is less well recognized is the damage this private life does to those who teach subjects early in a degree program, and which are prerequisite for many other "downstream" subjects. In the absence of a scholarly discourse on teaching, those who teach early in degree programs suffer the need to justify their assessment practices to those who teach later in the program whose assessment is rarely subjected to the same level of scrutiny.

In this paper, I have described and evaluated an approach to objective setting and assessment through multiple-choice examinations. Elements of the approach, such as the Bloom's taxonomy, are well grounded in education literature, as are the techniques used to analyze the multiple choice examination. Through a teaching discourse based on evidence and literature, the two lives of academics may eventually be united in a self-consistent whole.
Notes on Contributor

**Raymond Lister** is a senior lecturer, with 11 years of teaching experience, across three Australian universities, two of them ATN universities. His primary research interest is Computer Science Education. He is a past programme chair of the Australasian Computer Education Conference. He is also a regular columnist for the “Bulletin” of the international ACM Special Interest Group on Computer Science Education (SIGCSE). He has published 20 peer reviewed papers on aspects of Computer Science Education, nationally and internationally.

**Address for correspondence.** Faculty of Information Technology, University of Technology Sydney, PO Box 123 Broadway, NSW 2007, Australia. Email: raymond@it.uts.edu.au

**References**


Methods for Evaluating Multiple Choice Examinations


