

# Managing Large Class Assessment

Anthony Rhodes, Karyn Bower, Peter Bancroft  
Faculty of Information Technology  
Queensland University of Technology  
George Street, Brisbane 4000, Australia

[a.rhodes@qut.edu.au](mailto:a.rhodes@qut.edu.au)  
[k.bower@qut.edu.au](mailto:k.bower@qut.edu.au)  
[p.bancroft@qut.edu.au](mailto:p.bancroft@qut.edu.au)

## Abstract

ExamGen is a software application developed at Queensland University of Technology to assist instructors with assessment in large classes (250+). The ExamGen program was developed for the Faculty of Information Technology in response to an investigation into appropriate assessment alternatives designed to cope with the ever increasing enrolment numbers in tertiary institutions, in particular in Information Technology. This paper describes how ExamGen is being used to develop test banks of multiple choice and short answer questions from which examination papers can be generated to augment (or even largely replace) the current widespread use of traditional written theory exams. The advantages realised by this software tool include easy development and maintenance of unit test banks, automatic generation of examinations, faster marking and the ability to test a broader range of unit material. ExamGen has special features not found in similar software which assist instructors in maintaining questions which are not just straight recall.

*Keywords:* multiple choice questions, short answers, large class assessment support, examination generation software

## 1 Introduction

An increasing number of students are entering Australian universities. One of the problems that this brings for instructors is a subsequent increase in marking load. The problem has been compounded by the introduction of a summer semester in many Australian universities, which imposes further constraints, as the time between the end of one semester and the beginning of the next has diminished. One possible solution to this problem is to employ additional part-time markers. Unfortunately, suitably qualified tutors are not always available and this approach may introduce inconsistencies in marking, raising questions about the validity of the assessment. Tertiary instructors are forced to consider alternative forms of assessment instead of traditional, written

assignments and examinations. Electronic marking has become an increasingly popular solution to the problem but this restricts the types of questions that can be asked - multiple choice and short answer questions are amenable to electronic marking. It is often assumed that questions of this type are only able to test basic knowledge and understanding of concepts, however research suggests (Jones 1997, Kolstad 1994, Wilson 1991) that well written multiple choice and short answer questions can be used to test sub-synthesis cognitive complexity (Bloom 1956). The problem is that questions of this type are more difficult and time consuming to set.

Instructors who accept that multiple choice and short answer questions are a valid and effective means of assessment start to build up a collection of suitable material over time. Many instructors have accumulated their own test banks stored in a variety of forms, from plain text through to databases. The management of these stores often poses difficulties such as being able to add new questions, to allow sharing with other instructors, to generate examination papers, to ensure adequate coverage of topics and more importantly to provide facilities to create complex questions with additional features such as sequential inclusion or exclusion and graphics. The problem of effectively managing a collection of multiple choice and short answer questions gave rise to the ExamGen project which has resulted in a useful software product, currently utilised by instructors in the Faculty of Information Technology.

In (Lister 2000) it is recommended that multiple choice questions are effective for mid-semester examinations, where the focus is on the lower levels of cognitive complexity compared with final examinations. The ExamGen program has features that allow more complex questions to be created, testing the higher cognitive levels. For this reason, ExamGen is used by instructors at Queensland University of Technology for mid-semester tests and for up to one-third of the final examination, which helps to alleviate the problem of excessive marking in the time starved academic calendar.

In Section 2 we discuss the features of ExamGen, while in Section 3 we will illustrate its advantages and limitations and review the evaluation of the project. Section 4 compares ExamGen with other similar systems, and in Section 5 we consider the impact of ExamGen on future Teaching and Learning initiatives associated with assessing large classes.

## 2 ExamGen

ExamGen is a GUI-based, stand-alone, Java application, designed for the purpose of managing multiple choice and short answer questions stored in a Microsoft Access database. The database is linked to the application using Java-Database Connectivity (Horton 2000). Figure 1 shows the start up screen of the ExamGen program. The menus along the top bar provide a full set of functionality through the use of drop down menus and the most commonly accessed features are duplicated on a second menu bar using buttons.

Rather than describe the full functionality of ExamGen, the two most fundamental operations, adding questions and generating examinations will be briefly discussed. Then some of the special features will be described.

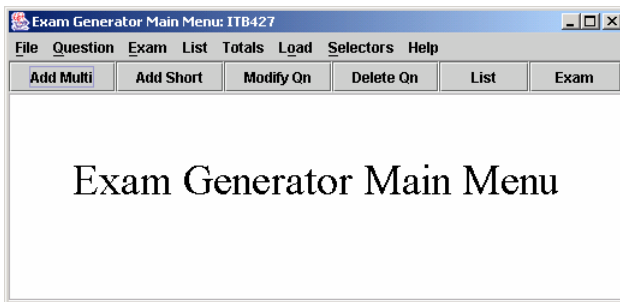


Figure 1: ExamGen Start Menu

### 2.1 Adding Questions

To add a multiple choice question, the instructor must first indicate to which topic it will be added and may then create a new question, as shown in Figure 2.

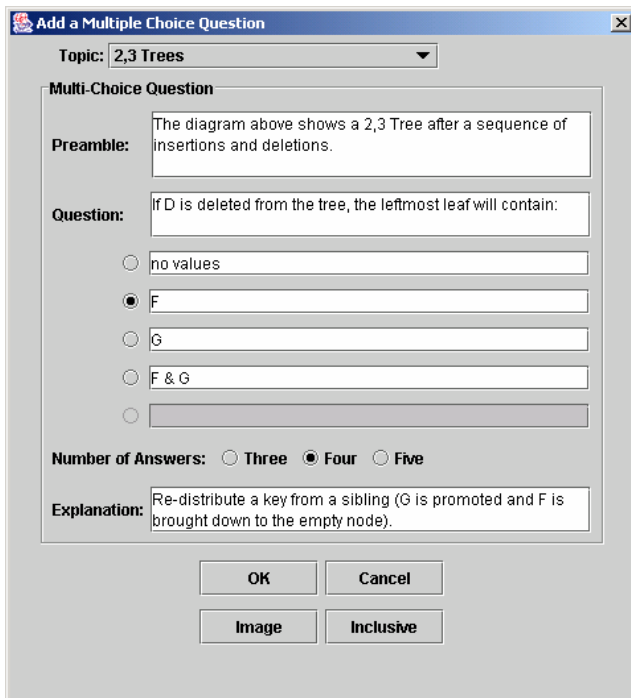


Figure 2: Adding a new multiple choice question

For each question the instructor may optionally write a preamble, as well as the actual question and between 3 and 5 choices. The correct answer is denoted by radio

button and an explanation for the answer may be included. Finally, the question is added (*OK* button) or cancelled. If a question is incomplete, ExamGen will provide a warning to indicate the missing details. The steps for adding a short answer question are similar.

### 2.2 The Examination Generation Process

Compared to similar tools, ExamGen provides good flexibility when generating an examination which may consist of multiple choice, short answer questions or a combination of both. Figure 3 shows the examination generation screen. It is possible to navigate through all of the questions for a particular topic, and choose only those questions you would like included. Navigation is achieved through the *Previous* and *Next* buttons at the bottom of the screen. The instructor may also elect to visit several or all topics, depending on whether the test is a weekly, mid-semester or end-semester examination. There is an option for displaying only the multiple choice or only the short answer questions if desired. An important feature of ExamGen is that it displays (in the title bar) the date that the currently viewed question was last used in an examination. When sufficient questions have been selected, the *Generate* button is selected.

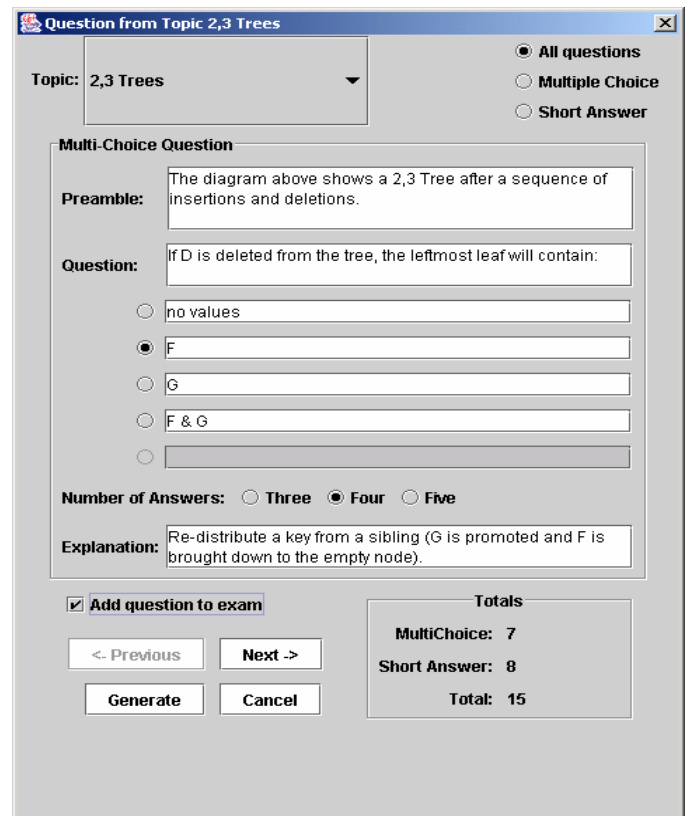


Figure 3: Generating an examination

The final activity in the examination generation process produces two files. The first is the questions file saved in html format. This can be loaded into a word processor for final editing by the instructor, e.g. adding a title, date or instructions to students. The second file is the answer file for the generated examination. In the case of multiple choice questions, this can be handed to the computer centre as a master answer sheet for the automated

marking of the student responses. For short answer questions, this becomes the marking guide.

## 2.3 Special Features

ExamGen provides special features which allow instructors to test higher levels of cognitive complexity rather than knowledge and comprehension, which are normally associated with multiple choice questions. For example, instructors are able to associate an image with a question, such as a diagram or piece of code (using the Image button on the form shown in Figure 2). When this graphic is associated with a well written question, the student can be required to perform tasks such as interpretation or classification of data, analysis of code and distinguishing of subtle differences in the operation of algorithms.

Another special feature of ExamGen is the ability to link questions through the use of mutual inclusion i.e. once the first question in a series is chosen for an examination, the others are automatically selected. It is difficult to set questions that test higher level cognition without providing a wordy preamble, but linked, sequential questions can be used to build on a concept, requiring students to solve a more complex problem at each stage. ExamGen also allows questions to be marked as mutually exclusive, ensuring that almost identical questions testing the same concept are not included on the same paper (mutual inclusion and exclusion are selected using the Inclusive button on the form shown in Figure 2).

ExamGen allows questions to be grouped using the *Selector* option. The groups are identified by a *Category Name*, which might be *Topic*, *Week*, *Chapter* or some other convenient classification. Once the category name is decided, the individual selectors are given meaningful names. For example, if the *Category Name* is *Week*, selectors such as *Week 1*, *Week 2* ... *Week 13* would be applied. Each new question is added to a selector corresponding to the week in the semester when that question is taught. In the examples above, *Topic* was the selector. This feature allows the instructors to group questions to suit their individual course requirements.

ExamGen allows the user to upload a batch of multiple choice or short answer questions from a text-based file in XML format.

```
<QUESTION_LIST>
  <QUESTION>
    <QUESTION_TEXT />
    <ANSWER />
    <ANSWER />
    <ANSWER />
    <ANSWER />
    <CORRECT />
    <EXPLANATION />
  </QUESTION>
</QUESTION_LIST>
```

This may be convenient for instructors who have existing material. Rather than having to retype all questions into ExamGen, the tags are inserted and the questions automatically loaded.

ExamGen shows a count of multiple choice and short answer questions for each selector. This allows the instructor to manage the balance between the two, while also revealing which selectors need more questions. It is also possible to produce a hard copy listing of all questions (or those belonging to particular selectors) in the testbank. Each question has a *Date last used* value, which makes the hardcopy listing very useful for browsing the testbank when choosing possible questions for the current test.

The features outlined in this section show that ExamGen is flexible, user-friendly and offers a wide range of functionality compared with similar programs.

## 3 Evaluation

A number of instructors in the Faculty of Information Technology have used ExamGen over the past year, and have provided feedback and suggestions for the improvement of the system. A summary of the main features of ExamGen that attracted comment is given.

- Significant time saved in generating examination papers, once the test bank has been created.
- Examinations are easily generated (in HTML format) with an accompanying answer file. The instructor need only add the examination heading and instruction for students, page breaks in appropriate places and to replace the links to images with the actual image.
- The date a question was last used is displayed, enabling the instructor to determine how much overlap there will be for the current and previous examinations.
- ExamGen has inclusive questions - selecting a base question automatically loads with it the associated questions.
- There is an optional preamble section for each question. This is very useful in IT for setting questions that have programming code blocks. It is also applicable to other disciplines where a long narrative is required (possibly for inclusive questions).
- Multiple choice questions can have 3, 4 or 5 options.
- ExamGen displays a count of questions for each selector and an overall count in the testbank. This is useful for quickly showing where more questions are needed.
- Deleted questions are flagged as inactive rather than being physically removed from the testbank. This enables later resurrection of such questions rather than having to re-enter all the question parameters.
- ExamGen has easy on-line searching for a question based on its ID (obtained from the hardcopy listing)

or a substring in the actual question (fast way to find and display all questions based on some keywords of interest to the instructor).

- Existing testbanks are easily loaded in XML format.

There were several suggestions for improvement:

- ExamGen is strictly a single user system and may cause problems with team teaching because concurrent access not supported. A better system for supporting team teaching would be a Web-based client-server system.
- Images are only stored as a link, therefore when the examination is generated, the instructor has to locate the image file and insert it into the question.
- Using ExamGen, multiple choice questions can only have 1 correct answer. Sometimes more than 1 correct answer may be required (although this is considered undesirable according to Kolstad et al 1990).
- Currently the installation of the system is manual and a little difficult even for IT literate people as it requires the instructor to be familiar with setting up a Java Database Connectivity link.
- External questions loaded through the XML conversion process described previously are not loaded with selectors; therefore each question must be edited to attach its selector.

A desirable outcome of using software such as ExamGen is the ability to test a wider and more comprehensive range of subject material in the examination. Rather than students being lucky enough to study just the right topics on a traditional written theory examination, they need to cover all the material if a test generated examination by ExamGen consists of say 60 multiple choice and 25 short answer questions. It is worth pointing out that the results obtained by students sitting mid-semester tests generated by ExamGen have proven to be in line with (in fact slightly less than) the final overall results for the unit. The following table shows the results for units which have used ExamGen.

Unit	Mid-Semester Average %	Overall Unit Average %
ITB469 2 <sup>nd</sup> Semester 2002	54	66
ITB427 2 <sup>nd</sup> Semester 2001	66	68
ITB427 1 <sup>st</sup> Semester 2002	66	71
ITB427 2 <sup>nd</sup> Semester 2002	64	66

**Table 1: Results of units using ExamGen**

A more thorough statistical analysis has not been attempted at this time due to the newness of the tool and a lack of sufficient test data.

#### 4 Related Work

Whilst a number of multiple choice generating systems already exist, e.g. HotPotatoes (Hot Potatoes 2003), TestPilot (TestPilot 2003) and Question Mark (Question Mark 2003), none of these systems fully meets the requirements of the instructors involved in the project. The following features are incorporated into ExamGen:

- link a graphic, or code fragment to questions
- produce a hard copy of an examination
- link related questions in either a mutually inclusive or exclusive way
- record when questions had been used on previous examinations
- track effectiveness of a question and its distractors

The ability to link related questions, either mutually inclusively or exclusively, was not present in any of the systems nor was the option of producing a written version of the exam. Question Mark was the only system to provide offline assessment without the need for a local server. Test Pilot was the only system that provided functionality to track the effectiveness of a question and its distractors, although Question Mark did offer services of this nature through their business partners. None of the systems kept track of the date the question was last used in an immediately accessible form.

The unique combination of features provided by ExamGen enables it to meet the needs of instructors who want an effective way of managing their test bank of questions.

#### 5 Conclusion and Future Work

ExamGen is a fully-operational, stand-alone Java application which is currently used in Queensland University of Technology's Faculty of Information Technology. Future work will concentrate on producing a web-based, client-server system that will encompass all of the features described in this paper and address the limitations outlined in the Evaluation Section, above. The most important enhancement to ExamGen will be concurrency support so that team teaching of units with multi-user access can be facilitated without introducing integrity problems. Drawing on the experience obtained from a recent web-based teaching and learning project, Environment for Learning to Program (ELP) ( Truong et al 2002, Truong et al 2003), the primary goal for ExamGen should be easily achieved and it is expected that the concurrent, client-server version will be integrated into the ELP. ExamGen will also continue to be a useful tool for instructors who use it in a stand-alone manner. With substantial testbanks (approximately 700 questions for ITB427 and ITB469) already developed it is an efficient means of producing multiple choice and short answer examinations and automatically generated solutions. Depending on the size of the examination,

instructors claim that this is certainly possible in less than one hour. Subsequent marking time as discussed above is very quick when compared to the time spent on a traditional written theory examination. The time invested in setting up a testbank for a subject is soon repaid. An added, but important, advantage is the ability to test a wider and more comprehensive range of subject material in an examination. Importantly, from a pedagogical viewpoint, examinations generated with ExamGen have proved to be able to challenge students' knowledge and understanding of unit material because of the system's capability for creating more complex questions.

## 6 Acknowledgements

The authors are thankful for the financial support for the ExamGen project which came from three principal sources in the Faculty of Information Technology at Queensland University of Technology - Teaching and Learning Committee, Technical Services Section and School of Software Engineering and Data Communications. We would also like to acknowledge the efforts of Queensland University of Technology graduate programmer, Mr Evan Dean for implementing ExamGen under our guidance.

## 7 References

Bloom, B.S. (ed.) (1956). Taxonomy of educational objectives. *Handbook 1. The cognitive domain* New York: David McKay.

Horton, I. (2000). *Beginning Java 2*. Wrox Press Ltd, Ch 19, 101.

Hot Potatoes (2003). Hot Potatoes – Half Baked Software. <http://web.uvic.ca/hrd/halfbaked/index.htm> Accessed 30 July 2003.

Jones, A. (1997). Setting objective tests. *Journal of Geography in Higher Education*, 21(1), 106.

Kolstad, R.K., Briggs, L.D. (1990) Multiple choice rules prevent the selection of wrong options in examinations. *Education*, 110(3), 360.

Kolstad, R.K., Kolstad, R.A. (1994). Applications of conventional and non-restrictive multiple-choice examination items. *Clearing House*, 67(6), 317.

Lister, R. (2000). On Blooming First Year Programming, and its Blooming Assessment. *Proceedings of the Fourth Australasian Computing Education Conference*, 4-6 December 2000, 158-162.

Question Mark (2003). Question Mark Home Page. <http://www.questionmark.com/us/home.htm> Accessed 30 July 2003.

Queensland University of Technology (2000). Teaching and Learning Grants and Awards. <http://www.talss.qut.edu.au/talss/stafftlgrants/gen/> Accessed 30 July 2003.

Test Pilot (2003). Test Pilot Online Assessment and Survey Engine. <http://www.clearlearning.com> Accessed 30 July 2003.

Truong, N., Bancroft, P., Roe, P. (2003). A Web Based Environment for Learning to Program. In *Proceedings of the 26<sup>th</sup> Annual Conference of ACSC*. January 2003.

Truong, N., Bancroft, P., Roe, P. (2002). ELP – A Web Environment for Learning to Program. In *Proceedings of the 19<sup>th</sup> Annual Conference of ASCILITE*. December 2002.

Wilson, T.L., Coyle, L. (1991). Improving multiple-choice questioning: Preparing students for standardized tests. *Clearing House*, 64(6), 422.