

A Methodology for Providing Individualised Computer-generated Feedback to Students

Michael Lambiris

Abstract: The traditional way of providing feedback to students after tests or assignments is labour-intensive. This paper explains the concepts and techniques used by the author to build computer-based applications that analyse students' answers and generate individualised, detailed and constructive feedback. The paper explains how the data gathered from a student's answers can be combined with other knowledge about the subject matter being taught, and the specific test questions, to create computerised routines that evaluate the individual student's performance. This information can be presented in ways that help students to assess their progress, both in relation to their acquired knowledge in specified areas of study, and with regard to their ability to exercise relevant skills. In this way, appropriate feedback can be provided to large numbers of students quickly and efficiently. The same techniques can be used to provide information to the instructor about the performance of the group as a whole, with a degree of detail and accuracy that exceeds the impressions usually gained through traditional marking. The paper also explains the role of the subject instructor in designing and creating feedback-generating applications. The methodologies described provide insight into the details of the process and are a useful basis for further experimentation and development.

Keywords: Teaching technology, computer-generated feedback, methodology and design, teaching large classes

1 Difficulties with Providing Good Feedback

It is widely recognised by educators that detailed, constructive, prompt and individualised feedback is an important aspect of good teaching and effective learning. See [1]. But providing feedback to students in the traditional form, that is, by reading the students' answers, evaluating them and writing comments, can be very time-consuming, especially with large classes. I teach a subject called Principles of Business Law that attracts enrolments of up to 700 students each semester. Assessment in this subject consists of four computerised tests, each comprising 30 to 40 multiple-choice questions. The tests are done under examination conditions. Scores are posted a day or two afterwards. It was in this context that I wished to provide individualised feedback after each test to the students. With classes of this size, it is impractical for an instructor to write comments for each student. A way was needed to produce feedback by means of a computer program.

2 What Should Feedback Consist of?

One way of providing feedback would be to publish the test questions together with the correct answers. This is often what students expect, but it may not be the best approach to learning. Thirty or forty questions cannot comprehensively test everything a student should know. A test is usually only a sampling of the student's knowledge and skills. When students correctly answer questions in a test, this indicates a probability that they know the relevant subject area well. Similarly, when they answer questions wrongly, this indicates a probability that they have an inadequate grasp of the subject area. If a student's answers demonstrate a weakness, they will likely need to revise that whole area of study rather than being given the correct answers to specific questions. Accordingly, my aim is to provide feedback in the form of general analysis, comment and advice. See [2].

3 Extracting Useful Information from Basic Data

Instructors who are knowledgeable in their own specialist area may not also be competent computer programmers and will need to employ specialist help to create application software. But the instructor needs to understand some basic programming concepts and techniques to be able to participate effectively in the process of designing and shaping feedback-generating software appropriately. In this paper I explain in detail one way in which such software can be created.

The starting point of is to identify what basic data is available. In each of the assessment tests that I use in teaching my subject, the students answer the questions by selecting a letter that represents their chosen answer (**a**, **b**, **c**, etc.). This letter is recorded in an electronic database so that the student's record consists of a sequence of 30 to 40 individual letters. A symbol (-) is used to indicate unanswered questions. A typical string of answers looks like this: **baabebccaecb-dabbab-abbceabaccaabaeadac**.

To create effective feedback, techniques are needed to extract more information from such basic data. How can this be done? Essentially, the process involves combining three types of information. The first is the particular answer the student chose for each question. The second is what the instructor knows generally about the subject area and skills being tested. The third is the focus or intent of each particular question in the test. The computer application can be designed to take proper account of these three factors and to draw specified conclusions from them. In this way, it is possible to build a useful picture of how the student has performed, and to identify their particular strengths and weaknesses. This information, properly presented with comments and advice, forms the basis for individualised feedback.

To fully envisage what is possible, it helps to understand the computer processes that are involved. An easy example of this is working out whether or not a particular answer is correct or incorrect. Essentially, the student's answer must be compared to the correct answer, to see if they are the same. A computer program compares data by using variables. Variables can be thought of as an electronic slots in which specified information can be stored. To compare a student's chosen answer with the correct answer, the data representing a student's answer can be retrieved from the database where it is permanently stored, and temporarily placed in a specified variable. The data representing the correct answer can be placed in another variable. The computer program then compares the contents of the two variables. If they match, it follows that the student has answered the question correctly and the **OK** result can be stored in a third variable, in the form of an increasing number or score. If there is no match, the student's answer is wrong and this **NOT OK** conclusion can be stored in a fourth variable (or by adjusting the number in the third variable downwards).

Using a process like this for all the student's answers, it is possible to work out how many answers were right or wrong. But a difficulty immediately emerges. It is apparent that the results obtained do not disclose *why* a student has chosen a particular answer. There can be many reasons for getting an answer wrong. For example, the student may have simply misread the question; or failed to understand the significance of a particular term used in the question; or not have had the necessary knowledge or skill to answer correctly. Similar possibilities exist in respect of correct answers. Taken individually, therefore, a student's correct and incorrect answers do not provide a sufficiently reliable basis for giving feedback and advice. But, given sufficient data, it is possible to look for significant patterns in a student's right and wrong answers. When all of the student's answers are analysed in the light of the particular knowledge and skills that the various questions are designed to test, distinct patterns emerge that can be useful as the basis for providing that student with helpful feedback.

4 Identifying Categories of Skill and Knowledge

To develop a computer-based application that carries out the necessary analysis of the student's answers requires careful thought and planning. The first step is to analyse each of the questions on the test, to identify, describe and name the particular categories of knowledge and skill involved. To do this

the instructor must combine their subject-matter expertise, teaching experience and examining skills. It may initially seem difficult to categorise each question in a specific and uncompromising way - some questions defy any neat classification. But, when classifying questions in one or more specified ways, it quite often happens that fresh insight is gained into what a question is truly attempting to do, and how that question might be improved so that it achieves its objectives more clearly and precisely. This is not a bad thing to happen.

Examples taken from specific tests illustrate the way in which categories may be defined. In the first test written by PBL students, analysis shows that each of the 40 questions involves one of three different generic skills. One is an ability to recall and apply acquired knowledge. Another is the ability to find specified information in a Statute and a Law Report. The third is the ability to understand, analyse, and draw conclusions from specific facts.

Each question can also be categorised according to the area of knowledge involved. In the test being discussed, the areas of knowledge are: (1) Constitutional arrangements and the organs of government in Australia; (2) The law-making powers of specified organs of government; (3) The processes and procedures for enacting legislation; (4) The hierarchy of the federal and state court systems; (5) The nature and organisation of law; (6) Understanding and appropriate use of legal terms and concepts; (7) The interpretation and application of statutory law; (8) The interpretation and application of case-law; and (9) Recognition and understanding of judicial reasoning.

When each category of skill and knowledge has been identified, it needs to be given a brief but distinctive name. Using the example above, the three categories of skill can be named **qt1**; **qt2** and **qt3**. The nine categories of knowledge can be named: **ch**; **lmp**; **nol**; **cs**; **lc**; **leg**; **sti**; **cl**; and **jr**. These names can be used (with some modification) to identify variables in a computer program. To carry out an analysis, the program will need three separate variables for each named category of skill and knowledge. This allows us to take account of whether the student answered that question rightly (r) or wrongly (w), or left it unanswered (n). Using this naming regime, the name of the first category above **qt1** is transformed into three variables named **qt1r**, **qt1w** or **qt1n**. Similarly, **qt2** becomes **qt2r**, **qt2w** and **qt2n**; and so on.

Further named variables will be needed to track other important aspects of the results, for example: **q1**; **q2**; **q3**, etc to hold the student's answer to each question; **right** to hold the correct answer being considered; **wrong** for the incorrect answer being considered; **result** for the result of comparing two variables; **rans** for the total number of correct answers; **wans** for the total number of incorrect answers; **nans** for the total number unanswered questions; **tans** for the total number of questions attempted; and **score** for the final score for the test.

5 Developing Routines for Analysis

To see what sort of information can now be extracted from the basic data requires some understanding of the computer-based processes involved. Imagine that we want to begin by analysing a student's answer to the first question in the test. The computer program begins by finding the particular student's string of answers in the database. It then selects the answer chosen by that student to the first question, and places the appropriate letter (a,b or c) in the relevant named variable, for example, in **q1**. Next, the program places the letter which represents the correct answer to that question (a, b or c) in the variable **right**. By comparing the letter stored in **q1** with the letter stored in **right**, the program can decide whether or not the question was correctly answered. This result can then be stored in a third variable where the total number of correct answers is kept: **rans**. If the question was not answered, this fact can be recorded in the variable that stores the total number of unanswered questions: **nans**. And if the question was answered wrongly, this conclusion is stored in the variable that stores the total number of incorrect answers: **wans**.

The program can now be made to classify the student's answer to the first question by reference to a category of skill. For example, assume question 1 tested the student's ability to understand, analyse, and draw conclusions from specific facts. Recall that the relevant variable for this skill was named **qt3**.

If the student got the answer right, the program can store this conclusion in the variable that counts the student's correct answers in this category - **qt3r**. Alternatively, if the question was answered wrongly, that conclusion can be recorded in the variable **qt3w** which shows the total of wrong answers in this category. Unanswered questions in this category are recorded in the variable **qt3n**.

The same procedure is followed to classify the student's answer to this question in relation to the area of knowledge being tested, using the variables **lmp_r**; **lmp_w**; or **lmp_n**. In this way, the student's answer to question one is being evaluated in various ways. The same routines are then repeated for each of the remaining questions, with appropriate changes to the variables used to store the conclusions. Once these basic routines have been carried out, further processes can be used to derive additional information from the data, or to organise it usefully. For example, the total number of questions answered by the student can be calculated by adding together the number of the student's correct and incorrect answers (**rans** + **wans**), and placing the result in the variable **tans** (for total answers).

Similar processes add to the value of the available information. So far, the individual questions have been classified as belonging to one of nine different areas of knowledge. For the purpose of generating feedback, the areas of knowledge can usefully be grouped into a smaller number of broader categories. The point of doing this is that it often helps students to understand where their strengths and weakness might lie in general terms, before going on to a more detailed analysis. In the first test written by PBL students, the nine areas of knowledge can be grouped into three broader categories, represented by variables **total1**, **total2** and **total3**, as shown below:

In **total1** the broad area of knowledge is **Organs, powers and processes of government** and includes: constitutional arrangements and the organs of government in Australia (**ch**); the law-making powers of specified organs of government (**lmp**); the processes and procedures for enacting legislation (**leg**); and the hierarchy of the federal and state court systems (**cs**). In **total2**, the broad area of knowledge is **Legal concepts and language** and includes: the nature and organisation of law (**no**); and understanding and appropriate use of legal terms and concepts (**lc**). In **total3** the broad area of knowledge is **The interpretation and application of law** and includes: the interpretation and application of statutory law (**sti**); the interpretation and application of case-law (**cl**); and recognition and understanding judicial reasoning (**jr**).

The totals in the relevant variables (shown in brackets above) are added together to show how the student has performed in each broad area of knowledge. This is done separately for right answers, wrong answers and unanswered questions. For example, the numbers in the variables **chr**; **lmp_r**; **leg_r**; and **csr** are added together in **total1_r** to show the correct answers in this broad area of knowledge, while **chw**; **lmp_w**; **leg_w**; and **csw** are added together in 'total1_w' to show the incorrect answers in this same area. The variables **chn**; **lmp_n**; **leg_n**; and **csn** are added together in **total1_n** to show the unanswered questions in this area. The same type of process can be used to produce data in relation to other specified learning objectives.

Finally, we can calculate the student's score for the test and place it in **score**. This is done by taking the number of correct answers (already contained in the variable **rans**) and doing whatever arithmetic calculation is needed to express it as a final mark. In the test now being discussed, a mark out of 15 is needed because the test counts for 15 per cent of the overall assessment for the subject. The number in **rans** is therefore divided by 2.667 and the result placed in **score**.

6 Presenting Information as Feedback

Using routines to analyse the basic data and extract additional information in the way described above is only the initial stage of actually providing feedback to a student. The next step is to build an interface that presents this data appropriately. The information available is sufficient to provide quite detailed feedback if it is built into a careful sequence of explanation, coupled with comment and advice. This should be presented in a clear, friendly, constructive and flexible way. One possibility is to follow

a traditional web-page design, with a list of contents on the left of the screen to indicate the extent and structure of the available feedback, with direct hyperlinks different sections. See figure 2 below.

As far as possible, the feedback should be individualised, by displaying the particular student's own data. In addition, particular comments and advice can be displayed selectively, depending on whether the particular student has a good score, an average score, or a poor score. The screenshots below provide examples. To script a full range of alternative comments and advice requires considerable forethought but the result is worthwhile. The feedback can also include information about how the individual student's performance compares to the class as a whole. And it can usefully include information and advice about future tests, for example, what new forms of question will be encountered, and what specific preparation may be needed. Students are very receptive to such information in the immediate aftermath of a test. The feedback applications can be made available to students either on a local area network, or by providing a downloadable version, or by running them on-line.

7 Providing Feedback to the Instructor

So far, this paper has been concerned with providing feedback to the students. But it is also important that the instructor get feedback on the effectiveness of their teaching, the validity of the questions set in the test, and the extent and accuracy of student learning. Traditional marking, which involves reading the answers, provides this feedback because, if a significant number of students make the same mistake, the instructor quickly becomes aware of the problem. With computer-based testing it is harder to get a clear idea of these matters. The normal output of a computer based test is a list of final marks and these do not tell the instructor much about where specific problems might lie. However, it is possible to use the techniques described above (with appropriate modification) to provide an analysis of the group results.

For a group analysis, the program begins by finding each student's string of answers in the database and carrying out the same sort of analysis already described, classifying the answers as right or wrong, and categorising the right and wrong answers in various ways, for example, by area of knowledge or skill, or in relation to specified learning objectives. As each student's string of answers is analysed, a cumulative total is built up, so that in the end it is known how many students in the entire group got each question right or wrong; what the distribution of marks is; what percentage of the answers were right or wrong in relation to particular areas of law; and what percentage of students satisfactorily demonstrated competency at particular skills.

This type of analysis would be time-consuming to do manually but it is quickly and easily accomplished using the methodologies described. The results give an accurate and clear picture of group performance - for example, see figures 6 and 7 below. If too many students appear to be answering a particular question wrongly, the instructor will quickly notice this and be able to investigate the different possibilities. It may be that the question is badly written; or that the topic is poorly taught; or that the student's have prepared inadequately in that area of study. Responding appropriately helps to improve the quality of the teaching and learning process.

8 Conclusions

By using appropriate techniques, and properly coordinating the skills and experience of instructors and computer-programmers, it is possible to automatically generate and deliver very satisfactory individualised feedback for students and instructors. Although the examples discussed here use the data obtained from computer-based tests in multiple-choice form, the same ideas could be adapted to tests that are not computer based, or that do not consist of multiple-choice questions. All that is required is to work out a marking scheme where numbers or letters are used to record the marker's evaluation of what the student has achieved. This data could be digitalised and used as the basis for computer-generated analysis

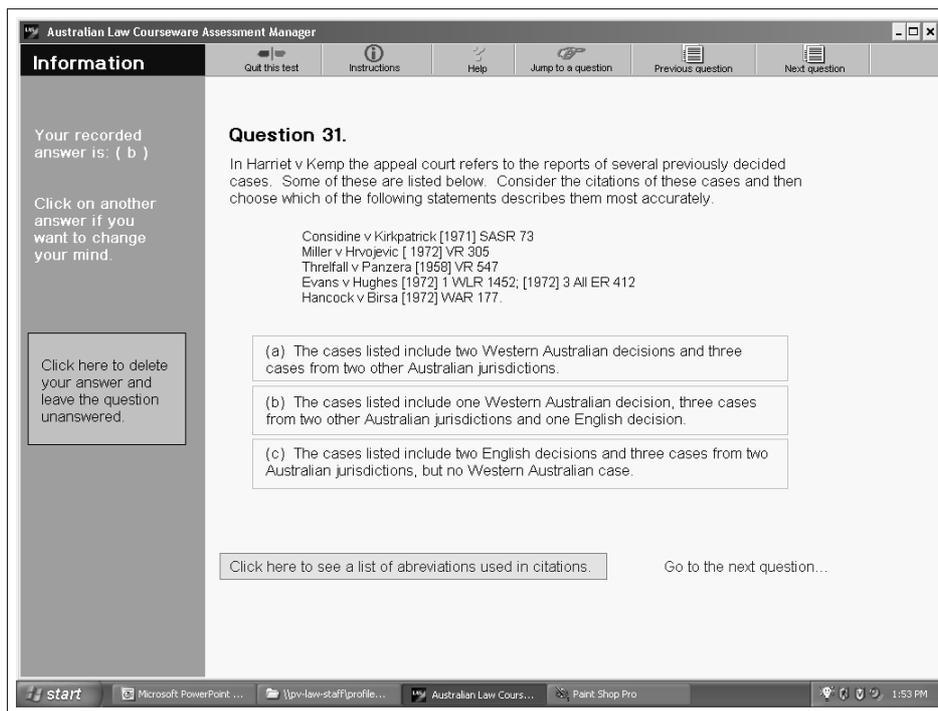


Figure 1: A sample question from a test. This question involves case-law, more specifically the meaning of coded information in case citations (variable **cl**). The student must interpret and evaluate the significance of that information (variable **qt3**).

and feedback, in much the same way as described in this paper. In essence, therefore, the techniques explained in this paper could find application in a wide range of situations.

The screenshots illustrate various aspects of the ideas explained in this paper. They show how the information generated from the basic data can be presented in a constructive, meaningful and readable style, and within a well-contextualised framework. The last two screenshots present an analysis of group data and show how a clear and detailed overview can be gained by the instructor of class performance as a whole.

References

- [1] Johnstone R, Patterson J and Rubinstein K, *Improving criteria and Feedback in Student Assessment in Law*, Cavendish Publishing, Australia, 1998.
- [2] East R, *Effective assessment strategies in law*, <http://www.ukcle.ac.uk/resources/assessment/effective.html>, 2005.
- [3] Higgins, E. and Tatham, L, *Assessing by multiple choice question (MCQ) tests*, <http://www.ukcle.ac.uk/resources/trns/mcqs/index.html>, 2003.
- [4] Lambiris M, *Assessment Management Software*, Australian Law Courseware Pty Ltd, Australia, <http://www.ALCware.com>, 2005 - 2006.

The screenshot shows a web application window titled "Australian Law Courseware Assessment Manager". The page is titled "Principles of Business Law" and "First Skills Task Feedback". On the left, a vertical list of 11 topics is shown, with "1. Your score and grade" selected. The main content area displays the following text:

1. Your score and grade

How your score was calculated. There were 40 questions in the test. You gained one mark for each correct answer. No marks were deducted for wrong answers or for questions left unanswered. Your final mark was converted to a mark out of 15 because the Skills Task counts for 15% of your final assessment in PBL.

Your recorded score is: 8.3 out of 15.
That mark gives you a P (Pass) grade.

The P grade represents quite a wide range of marks - between 7.5 and 9.6 out of 15. Your mark of 8.3 is in the middle of the grade. This represents a comfortable pass.

Provided you continue your present level of effort you are not in danger of failing. But there is plenty of room for improvement if you want to achieve better marks in future tests. This workbook will help you identify where you lost marks and how to do better in future tests.

Below the text is an illustration of two butterflies. At the bottom right, there is a link: "Go to the next section..."

Figure 2: In the feedback application, the topics are listed on the left of the screen with hyperlinks to the content of each section. This particular screen explains the scoring process, shows the individual student's final score and grade, and provides an appropriate comment.

The screenshot shows the same web application window, but the selected topic is "5.2. Organs, powers and processes of government". The main content area displays the following text:

5.2. Organs, powers and processes of government

There were twelve questions on how government in Australia is organised and how it operates. On these particular questions you got 5 answers right. You left 0 questions in this section unanswered.

Three questions tested your grasp of the constitutional arrangements and organs of government in the Australian Commonwealth, states and territories. You answered only one of these questions correctly. You must review the relevant materials.

There were **three** questions on the law-making powers of specified organs of government. You answered only one of these questions correctly. You must review the relevant materials.

There were **two** question on the processes and procedures for enacting legislation. You answered both of these questions correctly.

There were **four** questions on the hierarchy of the federal and state court systems. You answered only one of these questions correctly. You must review the relevant materials.

Overall, you have not got enough correct answers in this area of study. The subject matter is not difficult, so it is likely that you have not prepared in sufficient detail for the test. As soon as possible, you should go through the workbook 'Government and Law in Australia' again. Your experience in the test will help you focus on the important parts of that workbook.

At the bottom right, there is a link: "Continued on the next page..."

Figure 3: This screen provides a detailed analysis of the individual student's performance in a specified area of law (organs, powers and processes of government) and selectively provides appropriate comment. The feedback is based on the variables **total 1r; chr; lmp; leg** and **csr**.

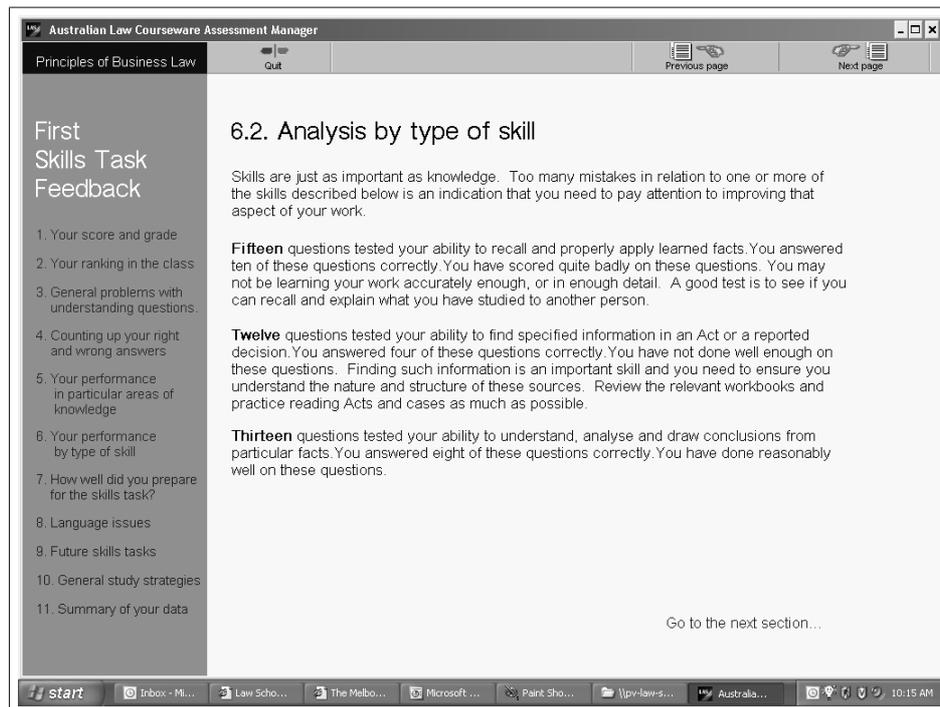


Figure 4: This screen uses the variables **qt1r**; **qt2r** and **qt3r** to analyse the individual student's ability to perform tasks involving specified skills. Appropriate comments are also displayed selectively, depending on the values in these variables.

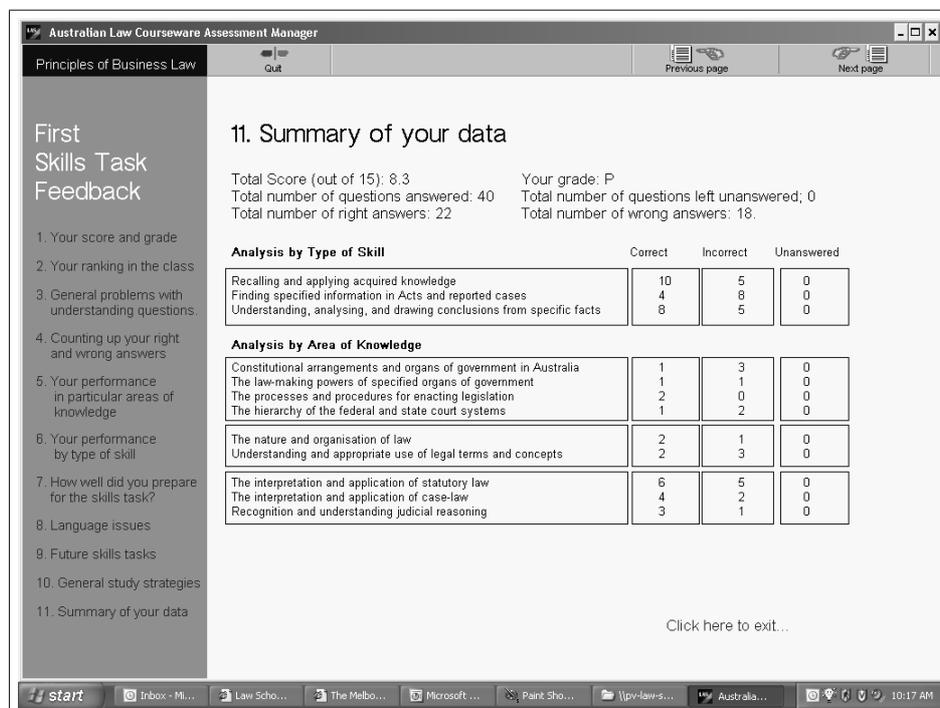


Figure 5: This screen summarises all of the available data. Presented in tabular form, it gives a concise overview of the student's performance. It also shows how a substantial amount of meaningful information can be generated from the basic data.

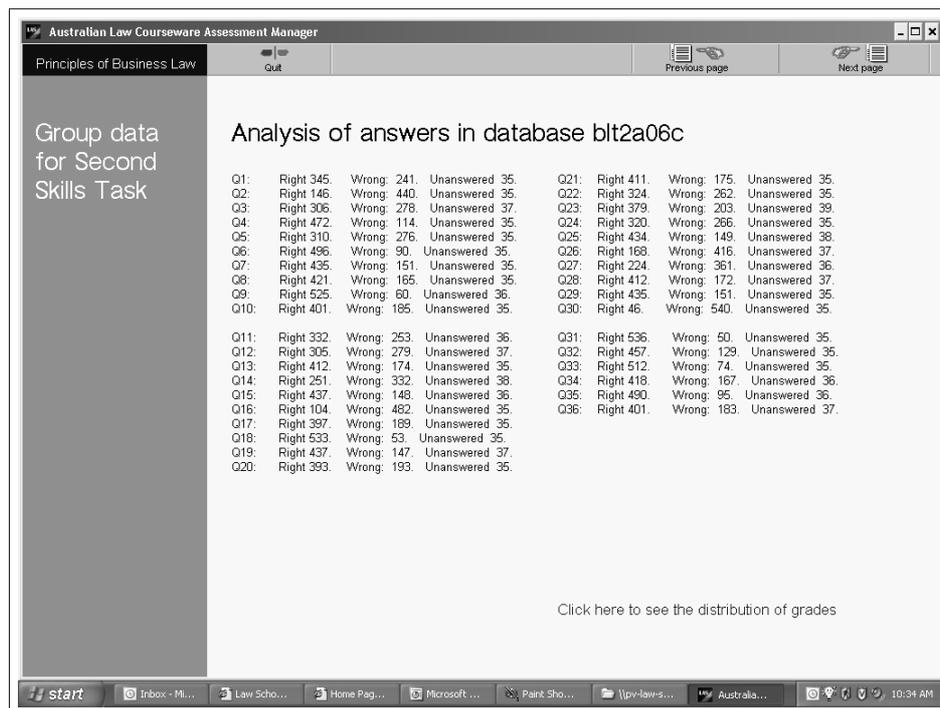


Figure 6: Using the same variables as devised for the feedback application, the data for the entire group of students can be generated for the instructor. This screen shows how many students in one group answered particular questions correctly or not.

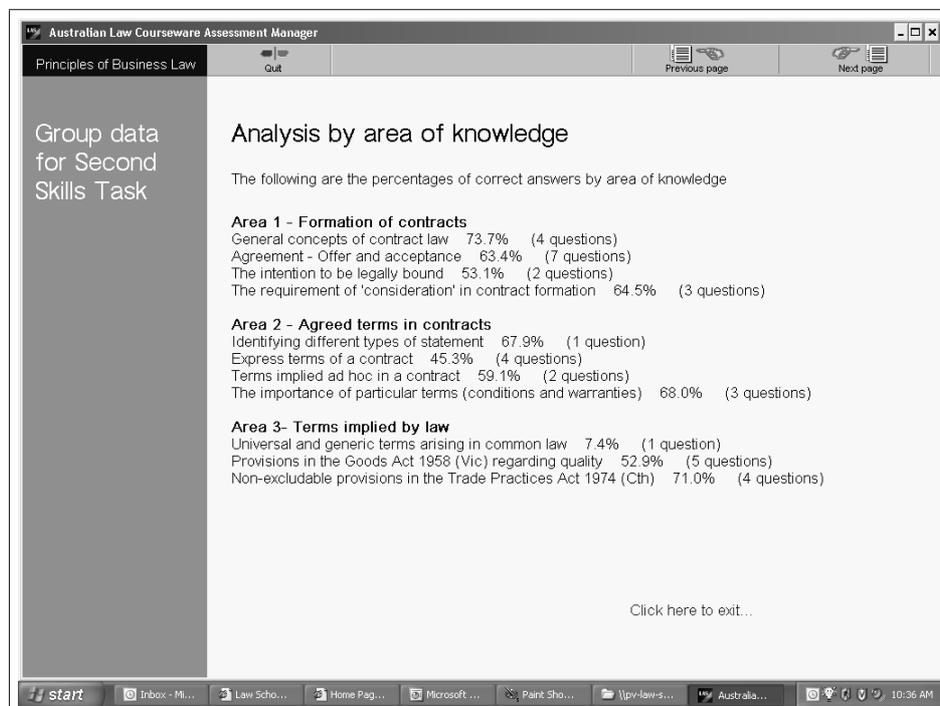


Figure 7: Group data can also give the instructor an overview of performance in relation to areas of knowledge, or particular skills. This screen shows the percentage of correct answers for the entire group in relation to the eleven areas of knowledge being tested.

Michael Lambiris
The University of Melbourne, Faculty of Law
Victoria 3010 Australia
E-mail: m.lambiris@unimelb.edu.au
Received: November 6, 2006

Editor's note about the author:



Michael LAMBIRIS (born January 22, 1950) obtained an LLB (Hons) from the University of London in 1971, and a PhD from Rhodes University in 1988. He has held positions at the University of Zimbabwe (1976-1982 and at Rhodes University, South Africa (1982-1991). He is presently an Associate Professor and Reader in the Faculty of Law, The University of Melbourne, Victoria, Australia. His main fields of teaching and research are commercial law and computer-based legal education. In addition to writing computer-based learning materials, he has developed computer-based testing and feedback software, written various papers and books and presented papers at many international conferences. He is the managing director of Australian Law Courseware (Pty) Ltd which publishes computer-based learning materials for law students.