Informatics in Education, 2004, Vol. 3, No. 2, 161–178

Testing Programming Skills with Multiple Choice Questions

David CLARK
School of Information Sciences&Engineering, University of Canberra, ACT 2601, Australia
e-mail: davidc@ise.canberra.edu.au

Received: July 2004

Abstract. Multiple choice questions are a convenient and popular means of testing beginning students in programming courses. However, they are qualitatively different from exam questions. This paper reports on a study into which types of multiple choice programming questions discriminate well on a final exam, and how well they predict exam scores.

Key words: multiple choice questions, testing, prediction, taxonomy, significance.

Introduction

Introductory programming subjects at universities can include students with a wide range of background. Some students may have taken programming subjects in secondary school, some may be self taught, whilst others may have no programming experience. Students with little or no programming experience, in particular, require early feedback, and it is usual to give one or more tests before the final exam at the end of the semester. These tests are often multiple choice, which have the advantage of automatic marking and therefore prompt feedback. However they have the disadvantages of allowing students who do not know an answer to guess, and of being qualitatively different from the questions in a final exam. With no other information to go on, students will extrapolate their test scores to the final exam. If the two are not related, students will have a false judgment of their progress.

This paper reports on a study aiming to investigate the relationship between results on multiple choice tests and the final exam. Specifically, this study had three major aims:

- to determine the significance of individual test questions;
- to determine which types of questions were significant;
- to find the subset of test questions that best predicted students' exam performances, and how good this prediction was.

The definition of significance is given below, but roughly it is how well the questions discriminated on students' performance in the final exams. There were two final exams, one to determine whether a student passed, and the other for higher grades. Test questions could be good discriminators on either, both or neither paper. A further aim was to see

whether test questions could discriminate between the credit level and the distinction/high distinction level scores on the final exam.

Once the significance of individual test questions are known, the effectiveness of types of questions can be determined. The categorisation was initially by type of question – tracing code, analysing code, etc. These types can be related to Bloom's cognitive levels. And the effect of level of difficulty can also be explored.

An assumption in this paper is that the results on the final exams are a good indicator of programming ability. In exams tasks can be set which require the same skills as are used in developing software. These include writing functions, finding errors and determining the purpose of code. Examples of exam questions are given in Appendix 1. Using students' assignments as the measure of programming ability would be desirable, but the help that students receive – both sanctioned and unsanctioned – precludes this. Exams, on the other hand, are entirely students' own work.

This paper describes the subject Introduction to Information Technology, relates multiple choice questions to the cognitive levels in Bloom's taxonomy, defines the measures used, discusses some factors which mitigated against the study, and finally reports the results and draws conclusions. A selection of the test questions, annotated, is given in Appendix 2.

Introduction to Information Technology

Introduction to Information Technology (IIT) is the introductory programming subject in the degree Bachelor of Information Technology at the University of Canberra. The students are not assumed to have any programming background. IIT is also part of a computing elective strand for students from other courses. It aims to be accessible to a range of students, whilst giving them a foundation for future programming subjects. The language of implementation is VB.NET. The final quarter of the subject gives an introduction to computer architecture. There are four assignments, the first three being programming assignments, each of which is followed immediately by a test with 15 questions. The questions are multiple choice, with four alternatives.

As well as the tests, there are two final exams, paper A and paper B. Paper A contains "easy" questions and is used only to determine whether a student passes or fails. Paper B is taken only by students who aspire to a higher grade. Having two final exams has proved very popular with the more moderate students. It was in part influenced by the paper of Pollard and Noble (2001).

The Role of Tests in Introduction to Information Technology

Although instructors prefer to think in terms of giving students feedback from tests and assignments, there is evidence that summative assessment (whose main purpose is grading) is the most important factor in determining the final learning outcomes for students (Crooks, 1988; Elton and Laurillard, 1979; Entwistle *et al.*, 1992). Students adapt their learning strategies to achieve good grades.

In IIT, therefore, the tests affect students' final grade. However the scores on the tests are not included in the totals for either paper. Instead, students who average 60% over all four tests have passed the subject and are not required to sit for paper A. But if they want a higher grade than a pass, they must still sit for paper B.

Continuous formative assessment is useful to students because it helps them understand their own learning progression. It gives them feedback on progress. The type of feedback will reinforce their learning behaviour and so it is important to give students feedback that most accurately reflects their progress. Continuous formative assessment is also of value to staff because it enables them to see how students are progressing and to adjust course materials and activities to counter difficulties that might arise. For this to be useful the feedback must be timely and meaningful. The first test in IIT is given in week 5 of a 15 week course.

Cognitive Levels and Question Types

Bloom's (Bloom (Ed.), 1956) well known taxonomy identifies six levels of cognitive skills, namely Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. Imrie's RECAP model (Imrie, 1984) adapts Bloom's taxonomy by dividing cognitive skills into two tiers. The first tier consists of the first three levels of Bloom's taxonomy namely Knowledge, Comprehension and Application but expresses them as the skills of Recall, Comprehension and Application. Bloom's final three levels of Analysis, Synthesis and Design are combined into a Problem-solving skills category. Imrie's simplified taxonomy can be more useful because of the close integration of all these skills in many computer-based problem-solving activities. Problem-solving often requires more than one skill and this broader classification removes the need for unnecessary distinctions in measuring skill performance. Achieving problem solving skills is the ultimate goal. Whilst they can be taught and assessed at an early stage (Cox and Clark, 1998), they are still dependent on the tier 1 skills in a particular subject.

When students first come to grips with a subject there is a tendency for knowledge and comprehension questions to dominate quizzes and for problem-solving to dominate later quizzes. However, it is possible for problem-solving questions to be set early in the course (Cox and Clark, 1998).

We now discuss and give examples of test questions at different cognitive levels.

Knowledge Questions

In early programming courses, the initial knowledge required is of the syntax of the language and the programming environment. By the end of the subject this knowledge is assumed and it is unlikely to be tested in a final exam. This is not to deprecate knowledge. Other cognitive skills depend on it. The answers to the question "what is the best way to solve this problem", for example, will depend on the students' knowledge of class libraries.

One or two simple knowledge questions can be useful in settling students into a test, especially early in the course. Question 13 is an example of a knowledge question.

Comprehension

Comprehension needs to follow quickly on the heels of knowledge. It consolidates knowledge and allows the internalization of knowledge. The easiest comprehension questions to construct are those where the students are required to trace code and find the final value of a variable. They can be used for selection statements, loops and arrays. Questions 3, 4, 7, 8, 9, 12 and 17 are examples.

Application

For a programmer, application is the ability to write programs and, at a finer grained level, to write a function or module to perform a specific task. This ability is learned in assignments and can best be tested in a standard written exam. However, "complete the code" multiple choice questions can be devised at this level. Question 10 is one such example.

Problem Solving

One of the most common tasks in maintenance is to work out what some code is doing. A popular type of question in a final exam is "what does this function do?" A variation is "under which conditions does this function perform correctly?" These tasks require the code to be analysed. Multiple choice questions which test analysis are not as easy to devise as comprehension questions, but with a little thought some of the tracing questions can be recast into analysis questions. Questions 11 and 17 is an example of such a pair. They were used as the first two questions in test 3, where it was hoped that the initial tracing would help the analysis. Question 5 is also an analysis question.

Although multiple choice evaluation and synthesis questions can be devised (Cox and Clark, 1998), they are difficult to construct in quantity and there were none in the tests used in this study.

Question Types and Partial Knowledge

Comprehension questions do not lend themselves to partial knowledge, and therefore eliminating some distracters. These questions are solved by tracing code and checking the answer from the list of alternatives. This is not true for analysis or application questions. For instance, in question 5 only 7% of students selected alternative B, and in question 10 only 10% of students selected options A or C. These seem to be distracters which are easily eliminated. Students who have successfully eliminated some of the distracters still have to guess, albeit with a greater chance of success. In cases such as these, a scoring system which rewards partial knowledge with partial marks reduces the effect of guessing (Clark and Pollard, 1989; Clark and Pollard, 2004). A simple way of doing this is to allow students to tick 2 alternatives and give them $\frac{1}{2}$mark if one of the ticked alternatives is correct.

Other Categorizations

Categorizing questions by their cognitive level provides useful insights, but other viewpoints can also be useful. Among these are questions relating directly to assignments and questions which test reflection. A further categorization is the level of difficulty of questions.

Assignment related questions have a context taken directly from an assignment. The implied question is "did you do your own work?" Examples are questions 1 and 2.

Reflection is defined by Dewey as *"the kind of thinking that consists in turning a subject over in the mind and giving it serious and consecutive consideration"* (Dewey, 1933, p. 18). Although testing reflection is best suited to assignments, reports and written exams, some assignment based questions are easier for students who have reflected on their assignment work. Question 6 is an example.

The level of difficulty is simply the proportion of correct responses. One of the aims of the study was to see if the level of difficulty had an effect on the effectiveness of the question. The questions in the Appendix 2 have a wide range of difficulty.

The Measures Used

Individual questions were evaluated using the significance measure. This is derived from the discrimination index as shown below.

discrimination $= (\mu_1 - \mu_2)/stdDev$
significance = *discrimination* / $\sqrt{1/n_1 + 1/n_2}$
$\mu_1(\mu_2)$ = *mean score for students who got the test question right* (*wrong*)
$n_1(n_2)$ = *number of students who got the test question right* (*wrong*)
$stdDev$ = *standard deviation for the test question*

The discrimination index measures the difference in scores between students who got the test question right and those who got it wrong. The standard deviation in the denominator makes it scale independent. The significance statistic is derived from the discrimination index. It takes the number of students into account. The more students, the more significant the results. A value of 2 or more is considered significant.

A particular test question's significance can be measured against scores for one or more questions on either paper. The total score is likely to be of most interest, but measuring against score on specific exam questions may provide some insights.

The N most significant test question are not necessarily the best N questions to predict students' results in the final exams. This is because test questions may be correlated. The correlation between questions is taken into account automatically by linear regression, and that is what is used in this study. Once the regression has been done, the students' final score can be predicted from the formula

$$\textit{Predicted score} = \textit{regression intercept} + \sum_i \textit{test question}_i \times \textit{regression coefficient}_i.$$

The effectiveness of the regression can be measured by how much of the variance in the data it explains. In the social sciences, a figure of 70% is considered reasonable. Although adding extra variables cannot increase the variance and will generally lower it, using all test questions is not necessarily the best option. Firstly, some of the regression coefficients may be negative, and whilst that may be improve the fit of the regression line it does not make sense in the context of using scores on test questions to predict exam scores. Secondly, a more appropriate statistic is the adjusted variance, which takes into account the number of variables. Typically, this rises to a maximum and then falls as more variables are added. Adding variables one at a time is known as step-wise regression and was used in this study to determine the best subset of questions to use.

Factors Mitigating against the Effectiveness of the Study

Some aspects of the tests and their role in IIT mitigated against the effectiveness of the study. Firstly the test questions had only 4 alternatives, and there were no penalties. Both of these increase the effect of guessing, and thus increase the level of noise in the data. Pollard and Clark (Clark and Pollard, 1989; Clark and Pollard, 2004) show that even 5 alternatives is an improvement over 4, and that the effect of guessing is dramatically reduced by having penalties for incorrect responses. Students dislike penalties and consider them unfair, so including them in determining test scores may be counter productive. Increasing the number of alternatives in at least some questions is not hard. In particular, students answer comprehension questions by working out the answer and checking that it is on the list. A new distracter does not have to correspond to a particular mistake. It just has to be in the right range.

The other factor mitigating against the effectiveness of the study was the way that the test scores were used. The idea was that students who averaged 60% on the tests would reach the end of the semester already having passed the subject. However, students who knew they could not make the 60% dropped out of the tests. Further, students who had averaged 60% did not sit for paper A and those who were content with a pass did not sit for paper B. The effect was that fewer weaker students were represented in the analyses involving tests 2 and 3, fewer stronger students were represented in the analyses involving paper A and fewer weaker students were represented in the analyses involving paper B.

Improving the Tests

The tests can be improved by making changes to the individual test questions, and in the way the tests are integrated into IIT. Examining questions with poor discrimination can identify problems with individual questions. Some question can be improved (question 4), whilst other questions may be better dropped (question 15) and new questions substituted. Even if the new questions are in the same category and of similar difficulty to an existing question, having several similar questions can mitigate against the effect of guessing. A further easy improvement is to increase the number of alternatives in some questions. This will reduce the effect of guessing. For some questions, 4 alternatives is natural (questions

6, 10, 12). For others, especially comprehension questions, adding an extra alternative is simple (questions 7, 8, 17). For instance "2" or "3" could be added to the alternatives in question 7. There is nothing to say that the number of alternatives must be the same on each question.

During the semester of this study, test scores were not incorporated into exam scores. The idea was to allow students to reach the end of the semester already having passed the subject. This has not worked out well in practise, and will not be continued. Instead, each test will contribute 5% of the marks on paper A. This decision has been made on pedagogical grounds, but it will have the effect of making the subset of students who do a particular test and a particular paper more typical of the students taking IIT.

Results of the Study

The broad results of the study were

- Knowledge questions were not significant for either paper.
- Almost all comprehension questions were good discriminators for paper B.
- Application questions were good discriminators for paper B, but not paper A.
- Analysis questions were good discriminators for paper B, but not for paper A.
- There is no overall relationship between the difficulty of a question and its significance.
- Very few questions were significant for paper A, and these tended to be either easy comprehension questions or questions relating directly to the assignment.
- The difficulty of the question may be a factor in how well a question discriminates between the harder questions on paper B.
- It is was not possible to make any useful predictions of the final score on paper A, but some can be made for paper B.

Knowledge questions. The only knowledge questions in the tests were the first two in the first test. They were intended to be easy questions to settle the students down. Their lack of significance was expected. Question 13 was one of them.

Comprehension questions. Almost all comprehension questions were good discriminators for paper B. They had a wide range of difficulty, from as low as 30% to as high as 85%. Questions 7 and 8 form an interesting pair. Question 8 seems a very small step up from question 7, but the proportion of correct responses dropped from 71% to 50%. Question 8 was significant for both papers, but question 7 was only significant for paper A.

Application questions. There was only one application question in the tests, number 10. That it was a good discriminator for paper B was pleasing. Such "complete the code" questions are not difficult to devise, and more will be added.

Analysis questions. It was expected that analysis questions would discriminate well for paper B but not for paper A, and this turned out to be the case. Questions 5 and 11 are

examples. Students generally found them difficult, which is not surprising as they are at a higher cognitive level. Care must be taken with analysis questions. It is easy to be too difficult (question 16) or to assume too much (question 15).

Level of difficulty. A pleasing outcome of the study is that questions do not have to be very difficult to be good discriminators for paper B. Question 9, which 74% of the students answered correctly, was a good discriminator for paper B as well as paper A. The best discriminator of all for paper B was question 12, which was answered correctly by 58% of students.

Paper A. The dearth of good discriminators for paper A was a disappointment. Contributing factors may include more guessing by weaker students, some stronger students not having to take paper A, and fewer weaker students taking the later tests. This latter point is supported by the fact that most of the good discriminators for paper A were on the first test. They fell into two categories: questions which related directly to the assignment just completed (questions 1, 2); and easy comprehension questions (questions 3, 7, 9). Question 7 is of interest in that it was taken directly from the lecture notes. Question 4 is also of interest. It is a simple loop trace, and was expected to discriminate well for paper A. That it did not may be due to the use of the "Step" syntax, which can be avoided by using a "While" loop.

At the top end. Two of the questions on paper B were intended for students aiming for a high distinction. Question 3 in Appendix 1 is one of them. There were a couple of test questions which were significant discriminators for these questions but not for easier questions on paper B. Both of the test questions were more difficult. Conversely a couple of the easier comprehension questions in the tests were not good discriminators on the harder questions in paper B.

Predictions. No subset of test questions could be found to make useful predictions on paper A, which is not surprising given the lack of questions which discriminate well on it. For paper B, linear regressions were done on questions from each test against paper B scores. The regressions on the three tests explained 69%, 64% and 74% of the variance respectively. In practise this means that students can be told that "most students who got your pattern of responses on this test got within 10% of this mark on the paper B". Whilst these are respectable rather than brilliant, they do give a better idea of students' progress than simply extrapolating the raw scores on the tests. A slightly better explanation of variance could be achieved by using all questions, but the stepwise regressions stopped adding new questions when the adjusted variance, which takes into account the number of variables, was maximised. Also, including all questions in the regressions made some coefficients negative. Although this may improve the regression fit, it does not make sense to say to a student "getting this question right will lower your predicted score on the final exam".

Conclusions

This study has provided insights into the type of questions which are good discriminators for moderate students who are concerned with passing and for stronger students who aspire to higher grades.

No single form of assessment is sufficient for all learning outcomes, and multiple choice tests in particular have their weaknesses. However the tests in this study are useful in predicting performance in the final exam. This gives confidence in their role in summative assessment, and it is anticipated that changes already made in the structure of assessment in IIT will improve the tests' predictive ability.

Appendix 1: Sample Exam Questions

This appendix contains a selection of questions used in the exams, with a brief comment on each.

Question 1

The array *data* contains the following values.

4	4	5	5	6	6	6	7	7	7	8	8

The variable *numItems* contains 12.

i) What is the value of *count* after the following code fragment is executed?

ii) State in one sentence what the code does.

```
count = 0
For i = 1 to numItems - 1
  If (data(i) = data(i - 1)) Then
    count = count + 1
  End if
Next i
```

Comment. A question on paper A. The tracing was intended to help the analysis.

Question 2

Write in VB a Boolean function which will take a string as parameter and return True if the string is a palindrome and False if it is not. A palindrome is a word which reads the same backwards as forwards. Examples of palindromes are *noon* and *madam*.

You may assume that the all of the characters in the string are lower case letters (no spaces).

Recall that if the string *str* contains "`hello`" then *str.Length* is 5 and *str.Chars*(1) is '`e`'.

Comment. One of several parts of a coding question on paper B.

Question 3

A report is to be written with *maxLines* lines on each page, except possibly the last. Each page is to have a header which is written by the sub *writeHeader*(*header As String*). Lines are written by the sub *writeLine*(*line As String*). There are no footers on this report.

The following algorithm attempts to write the report. (Assume that correct declarations have been made.)

```
writeHeader(header)
lineNo = 0
For i = 0 To n-1
    lineNo = lineNo + 1
    If (lineNo >= maxLines) Then
       writeHeader(header)
       lineNo = 0
    End If
    writeLine(lines(i))
Next i
```

The algorithm does not do what is intended. Describe what it does wrong.

Hint. Work out what happens if *maxLines* is 3 and n is 10.

Comment. One of the more difficult questions in paper B. Writing a report was part of the 3rd assignment, and students used an algorithm similar to this.

Appendix 2: Test Questions

This appendix contains a selection of questions used in the tests, with a brief comment on each. The appendix contains "discriminating" questions whose significance measure was greater than 2 for one or both papers, and "non-discriminating" questions whose significance measure was less than 2 on both papers. The numbers in parentheses are % correct, significance on paper A and significance on paper B.

Questions which discriminated well on at least one paper

Question 1

In calculating the cost of a number of tickets, which of the following is NOT used?

A The `SelectedItem` property of a list box.
C The `Checked` property of a radio button.
B The `SelectedIndex` property of a list box.
D The `Text` property of a text box.

Comment. One of the few questions significant for both papers. It relates to the first assignment (70%; 3.1; 3,6).

Question 2

The code

```
label1.Text = "$" & x
label2.Text = FormatCurrency(x)
```

is executed 3 times, with `x = 123.45, x = 123.456` and `x = 1234`. For how many of these values of `x` are `label1.Text` and `label2.Text` the same?

A 0
C 2
B 1
D 3

Comment. "FormatCurrency" was used in the first assignment. The 1234 was an error. It should have been 123.4 (46%; 3.8; 3.6).

Question 3

What is the value of m after the following code fragment is executed?

```
a = 2
b = 2
c = 4
If (a > b) Then
    If (a > c) Then
        m = 1
    Else
        m = 2
    End If
Else
    If (c <> 0) Then
        m = 3
    Else
        m = 4
    End If
End If
```

A 1
C 2
B 3
D 4

Comment. A simple loop tracing exercise. Surprisingly effective (85%; 2.3; 1.8).

Question 4

How many times will the following loop be executed?

```
For i = 2 To 14 Step 3
    '  body of loop
Next i
```

A 4
C 12
B 5
D 13

Comment. Intended as an easy question. Significant in paper B, but not paper A. Option D was the most popular distracter. Weaker students may be unfamiliar with the "Step" syntax (51%; 0.0; 4.1).

Question 5

Which is true after the following code is executed? Assume that x is >= 0.

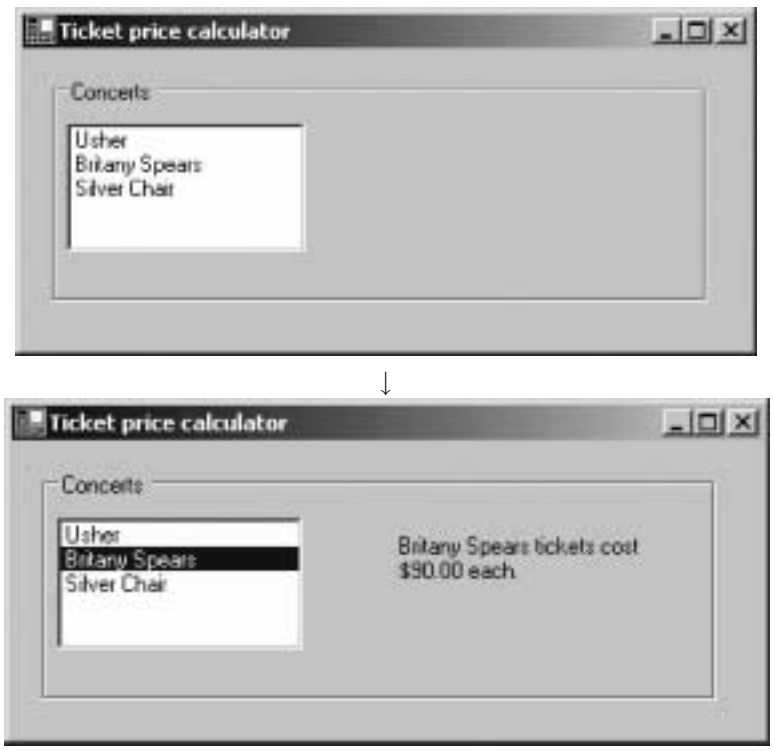
```
y = 0
Do While (y < x)
      y = y + 7
Loop
```

A y must be greater than x.
B y may equal $x + 7$.
C y must be greater than 0.
D y may be equal to x.

Comment. An analysis question. As expected, effective at the higher level but not at the lower level. Option C was the most popular distracter (50%; 0.9; 5.2).

Question 6

In assignment 1 when a concert is selected the cost of each ticket is displayed.



Assume that the performer and ticket price are stored in arrays. In the code which sets the label's Text property,

A neither a selection (IF statement) nor a loop is necessary.
B a selection is necessary but a loop is not.
C a loop is necessary but a selection is not.
D both a selection and a loop are necessary.

Comment. Relates directly to the first assignment. Weaker students may have been confused by the use of "selection" as they would have used the "SelectedIndex" property of the ListBox control. Better to have just used "IF statement". It may also favour students who have reflected on their work (27%; 0.0; 3.8).

Question 7

The variable *count* has the value 6, and the array x contains the following values:

23	12	13	17	23	19

What is the value of m after the following code is executed?

```
m = 0
For i = 0 To count - 1
    If (x(i) > x(m)) Then
        m = i
    End If
Next i
```

A 0
B 1
C 4
D 5

Comment. A simple tracing exercise using arrays. The example was discussed in lectures, so should have been familiar to the students. Effective for both papers (71%; 2.8; 5.1).

Question 8

The variable *count* has the value 6, and the array x contains the following values:

1	3	0	2	2	4

What is the value of p after the following code is executed?

```
p = 0
For i = 1 To count - 1
    If (x(i) > x(i-1)) Then
        p = p + 1
    End If
Next i
```

A 1
B 2
C 3
D 4

Comment. Appears to be only a small step from question 8, but weaker students found it difficult (50%; 0.8; 6.8).

Question 9

Which of the following code fragments produces this output (*sw* is a stream writer)?

```
1234
234
```

```
34
4
```

A
```
For i = 1 To 4
    For j = 1 To i
      sw.write(i)
    Next j
    sw.writeLine()
Next i
```

C
```
For i = 1 To 4
    For j = i To 4
        sw.write(i)
    Next j
    sw.writeLine()
Next i
```

B
```
For i = 1 To 4
    For j = 1 To i
        sw.write(j)
    Next j
    sw.writeLine()
Next i
```

D
```
For i = 1 To 4
    For j = i To 4
        sw.write(j)
    Next j
    sw.writeLine()
Next i
```

Comment. A popular question for many years. It's effectiveness is gratifying (74%; 2.1; 3.9).

Question 10

The array *data* contains *numItems* elements. The code below is intended to shift the elements in *data* to the left with wrap around. That is to convert

7	3	8	1	0	5

into

3	8	1	0	5	7

```
"statement 1"
For i = 0 to numItems - 2
    data(i) = data(i+1)
Next i
"statement 2"
```

In order for the code to execute correctly "statement 1" and "statement 2" should be

A
```
temp = data(0)
data(0) = temp
```

C
```
temp = data(numItems-1)
data(0) = temp
```

B
```
temp = data(0)
data(numItems-1) = temp
```

D
```
temp = data(numItems-1)
data(numItems-1) = temp
```

Comment. The only "coding" question in the tests. That it is effective is pleasing (76%; 1.2; 2.9).

Question 11

Assuming that x is >= 0, which of the statements below is **false** after the code is executed?

```
y = 1
Do While (y <= x)
    y = y * 2
```

```
Loop
```

A y **may** be odd.
B y **may** be equal to x.
C y **may** be equal to $x + 1$.
D y **must** be a power of 2.

Comment. An analysis question. Option A was the popular distracter. This was given as the paired with question 17 as the first two questions of the 3rd test (39%; 1.4; 4.7).

Question 12

The function Z takes a string as an argument and returns a Boolean value (True or False).

```
Function Z(ByVal pString As String) As Boolean
    Dim i, j As Integer
    Dim response As Boolean
    i = 0
    j = pString.Length - 1
    response = True
    Do While (i < j)
        If (pString.Chars(i) <> pString.Chars(j)) Then
            response = False
        End If
        i = i + 1
        j = j - 1
    Loop
    Return response
End Function
```

The code below is run, using the function Z above.

```
Dim noon, madam As Boolean
noon = Z(“noon”)
madam = Z(“madam”)
```

Choose the correct option.

A After the code is executed *noon* is True and *madam* is False.
B After the code is executed *noon* is False and *madam* is True.
C After the code is executed *noon* and *madam* are both True.
D After the code is executed *noon* and *madam* are both False.

Comment. The highest significance of any question in any test. May be solved by tracing or analysis (58%; 1.9; 8.0).

Questions which did not discriminate well on either paper

Question 13

In Windows 2000 or Windows NT, right-clicking on an object on the desktop or in a folder has what effect?

A No effect. Windows does not use the right mouse button.

B Pops up a Context menu at the point you clicked, showing the properties of the desktop or folder in which the object resides.

C It invokes the Start menu.

D Pops up a Context menu at the point you clicked, showing the main methods of the object, and an item for its properties.

Comment. A simple knowledge question, used as to settle the students down in their first test. Not significant in either paper. (81%; 1.4; 1.1)

Question 14

Which of the following could NOT be used to allow a user to indicate whether a person was male or female.

A A single check box.

B A list box with two items (“male” and “female”).

C A pair of option buttons.

D None of the above – any of them could be used.

Comment. An attempt at an analysis question on controls rather than code (84%; 1.1; 1.6).

Question 15

The following code attempts to count how many even numbers there are between m and n inclusive (Assume that $m <= n.$).

```
evens = 0
For i = m to n Step 2
    evens = evens + 1
Next i
```

Which of the following is true?

A The code succeeds for all values of m and n.

B The code fails if m is odd and n is even.

C The code fails if m is even and n is odd.

D The code fails if m and n are both odd.

Comment. A bad question! Intended as an analysis question but may assume more comfort with mathematics than is typical for IIT students (42%; 0.0; 0.6).

Question 16

The following algorithm is intended to set the variable *range* to the difference between the largest element and smallest elements in the array *data*. For example in the array below it attempts to set *range* to 7 (9−2).

6	3	8	9	7	2

Assume that *numItems*, the number of elements in the *data* is 1 or more, and that the following code fragment has been executed.

```
x = 0
y = 0
For i = 1 to numItems - 1
```

```
        If (data(i) > data(x)) Then
            x = i
        End If
        If (data(i) < data(y)) Then
            y = i
        End If
    Next i
    range = data(x) - data(y)
```

Which of the following is true? If you believe that more than one option is true, choose the first correct option.

A The algorithm fails if all of the elements are the same.

B The algorithm fails if the largest element is the first in the array.

C The algorithm fails if *numItems* is 1.

D The algorithm is correct for any data provided *numItems* is 1 or more.

Comment. An analysis question, but too difficult for the majority of students (22%; 0.4; -0.5).

Question 17

If x is 8, what is the value of y after the code is executed?

```
y = 1
Do While (y <= x)
    y = y * 2
Loop
```

A 8

B 10

C 12

D 16

Comment. The first of a pair of questions using the same code. This was the comprehension question. The second was the analysis question number 11. It was set in test 3, which was not attempted by some of the weaker students (71%; 0.8; 1.3).

References

Bloom, B.S. (Ed.) (1956). *Taxonomy of Educational Objectives*: *Handbook 1, Cognitive Domain*. New York, Longman.

Clark, D.I., and G.H. Pollard (1989). An optimal scoring system for multiple choice competitions. *J. World Fed. Nat. Math. Competitions 2*, **2**, 33–36.

Clark, D.I., and G.H. Pollard (2004). A measure of the effectiveness of multiple choice tests in the presence of guessing: Part 1, Crisp Knowledge. *J. World Fed. Nat. Math. Competitions*, **17** (1), 17–64.

Cox, K., and D. Clark (1998). The use of formative quizzes for deep learning in programming courses. *Computers in Education*, **30**(3/4), 157–167.

Crooks, T.J. (1988). The impact of evaluation practices on students. *Review of Educational Research*, **58**, 438–481.

Dewey, J. (1933). *How We Think*. D.C. Heath, Boston.

Elton, L., and D. Laurillard (1979). Trends in student learning. *Studies in Higher Education*, **4**, 87–102.

Entwistle, N., S. Thompson and H. Tait (1992). *Guidelines for Promoting Effective Learning*. Centre for Learning and Instruction, University of Edinburgh, Edinburgh.

Imrie, B.W. (1984). In search of academic excellence: samples of experience. In *Proceedings of the Tenth International Conference on Improving University Experience*. University of Maryland, University College, pp. 160–183.

Pollard, G., and K. Noble (2001). The increased reliability and efficiency of dynamic assessment methods. *Mathematics Competitions*, **14**(2).

D. Clark completed his PhD from ANU in 1981. He has taught for two years at the University of Kentucky, and for 20 years at the University of Canberra. His research interests include teaching, testing and neural networks. He has been a member of the committee of the Australian Mathematics Foundation since 1985, and a member of the Australian Informatics Olympiad since its inception in 1999. He attended the International Olympiads in Informatics in 2001, 2002 and 2003 as Australian deputy team leader.

Programavimo įgūdžių tikrinimas naudojant pasirenkamųjų atsakymų testus

David CLARK

Testai, kuriuose iš kelių pateiktų atsakymų pasirenkamas vienas, yra patogi ir populiari priemonė pradedančiųjų studentų programavimo žinioms patikrinti. Vis dėlto, testo ir egzamino klausimai kokybiškai skiriasi. Straipsnyje aptariami programavimo žinioms patikrinti skirtų testų tipai, nagrinėjama, kiek tokie testai veiksmingi diferencijuojant studentų žinias bei kiek šių testų rezultatai gali atspindėti baigiamojo egzamino rezultatus. Siekiant nustatyti ar studento žinios atitinka patenkinamą lygį, pravartu pateikti klausimus programos teksto analizei ir klausimus, kurie tiesiogiai susiję su paskaitų metu pateikta medžiaga. Dirbant sudėtingesniu lygiu studentų žinias gerai diferencijuoja programos teksto analizei skirti klausimai bei nurodymai užbaigti ar papildyti programas. Straipsnyje pateikiami visų klausimų tipų pavyzdžiai, aptariami ir mažiau naudingi studentų žinių diferencijavimo požiūriu klausimai. Baigiama išvada, kad vertinant ir diferencijuojant studentų žinias apgalvotai parinkti klausimai yra pravartesni nei vien paprasti testai.