Flexible Assessment for Programme Development, PD100

Oliver Hoffman
Manukau Institute of Technology
Auckland, New Zealand
oliver.hoffman@manukau.ac.nz

ABSTRACT

Tests favour quick thinkers. They place others under a certain amount of pressure. Performance is frequently adversely affected. Assignments are difficult to control: Does the work represent an original effort?

Normally we set two or three assessment items per course. This means students need to rely on formative assessment in order to obtain feedback on their progress. It also means that the summative assessment can be a heavy burden on them.

At MIT in the past we have used both tests and assignments to assess in particular the application work in Program Development (PD100). Last year we trialled an innovative approach using flexible ongoing assessment. In each session there was either a short test (of 15 minutes duration) or a homework task. In all, eighteen items were offered: six tests and twelve exercises. Each one was worth ten marks (or 10% of the final mark). They were spread out over the whole course and ultimately the ten best marks made up the overall total mark.

The results were very encouraging. Students worked harder. They had an ongoing measure of progress. There were no surprises at the end. The pass rate was higher. And non-original work could be detected early. It is intended to continue with this method and to apply it to other courses - in particular to Programming Principles - in future.

1. INTRODUCTION

After teaching PD100 for several semesters I decided on a new approach to the assessment for the first semester of the year 2000. In the past I had used one test and one assignment. Problems had arisen when the assignment - usually of moderate size - had been left until the last minute or, worse, when students had not been able to resist the temptation of working together too closely. In the second semester of 1999 I used two tests, but found that a number of students were running into time trouble when it came to preparing problem solutions and structure diagrams within the given timeframe.

In 2000 I therefore introduced an ongoing flexible assessment approach. This was explained in the course outline, the relevant parts of which are in appendix A. In summary, there were four mini-tests (in lessons 3 to 6), twelve exercises or mini-assignments (lessons 7 to 14), and two mini-tests at the end (lesson 15). In addition to that, I allowed one resit test and one resit exercise in week 17 of the semester. The resits were selected in areas where the student had missed an earlier test or exercise or where the student had performed poorly. This was decided on an individual basis.

2. RESULTS

The results showed a higher than normal pass rate by about 10 % (75 out of 96 passed). This is certainly due in part to the higher level of work put into the subject by the students and probably also due to the higher level of motivation created by the flexible assessment approach. It cannot be denied that maybe about five students ended up passing the course when they would almost certainly not have done so under previously used assessment methods. However it is recognised that they deserved their pass because of the extra effort that the assessment method encouraged them to adopt; at the same time, they would have needed to continue that effort in order to be equally successful with Programming Principles.

3. ADVANTAGES

Three main advantages emerged:

- Material was learned immediately after it was taught. Students were assessed on the material while it was fresh in their minds.
- There was less pressure on students on two counts: firstly on a general level throughout the course and secondly at what would otherwise have been the critical times of the course (based around the times of major tests and assignments).
- Students actually worked harder than they had in the past. This is because they effectively completed 12 mini-assignments (in addition to the tests), achieving a possible total of 180 marks. In the past 100 marks would have been sufficient to assess the course. The majority of students did far more assessed work than they needed to have done in order to pass the course.

4. DISADVANTAGES

In this first trial there were two disadvantages which I was able to eliminate in subsequent re-runs of the assessment method. There was also one feature that may or may not be a disadvantage.

With the assessment design used on this first trial run, it was possible for students to pass the course about one or two sessions before the completion of the course. A few students - about 15% - therefore elected not to attend the last class; a larger number opted not to sit either of the two mini-tests (5 and 6) associated with that class. While this in itself is not a major disaster (the last session covered an overview of programming languages), it does not represent a desirable goal. In later semesters an enhancement was added which encouraged full participation until the end. One of the last two mini-tests was declared to be a compulsory one.

In an effort to introduce a reasonable amount of flexibility there were four sessions where a student could elect to work on two exercises instead of the usual one. The idea was for students to hand in the one exercise they felt more comfortable with. Instead, the majority of students opted to do both exercises. With the actual scoring system used, it would have been better for them to do one exercise well (and score 9 or 10 out of 10) rather than to do two exercises and score only average (7 or 8). This actually added to their workload but at the same time it gave them the opportunity to familiarise themselves with two different aspects of the course work covered and was therefore a worthwhile additional learning experience.

From the tutor's point of view there was probably overall more marking to do (but spread out throughout the semester) and, on this first run, certainly more work in preparing the various assessment items.

A point for discussion concerns the issue of retention. If students learn the material and are assessed on it immediately after it has been taught, will the retention factor be higher or lower than if they learn all the material at once for a big test? No clear answer to this question emerged from this trial run.

5. COLLUSION

A major problem in the assessment of PD100 in the past has revolved around the issue of collusion . The temptation to work as a “team” when doing assignments has often proved to be irresistible. While collusion on the mini-exercises is still possible and was indeed detected in about five cases, the damage was less and it was not necessary for a student to have to repeat a whole assignment. With such cases being picked up early on, an appropriate message could be delivered before the damage became too great. To avoid comparison of tests between streams a different test was used for each stream. Ideally a selection of exercises could be offered as well but it will take time to build up a larger exercise assessment bank.

6. PROGRESS IN SUBSEQUENT TRIALS

To maintain student interest and energy up until the end of the course one of the last two tests was made a compulsory one (students' choice). This was indeed sufficient to stimulate the interest and level of commitment right up till the last lesson.

To reduce the workload somewhat (for both the student and the marker!) a reduction in the number of assessment items was made. Hand in hand with the change in the focus of the syllabus the number of tests was reduced from six to five; at the same time the number of exercises was reduced from twelve to nine. To compensate for the reduction two additional items were allowed at the end of the course. These served as resits for those students whose marks after 14 assessment items had not yet reached the pass level. Where a student had missed an earlier exercise or test due to sickness or excused absence a student was permitted to hand in answers to an additional exercise. Three such additional exercises were offered.

7. STUDENT FEEDBACK

Student evaluation was carried out at the end of the course and a summary of the results is shown in appendix B. Obviously the method was well received by the students. Many of the areas where improvements were indicated were covered by the above proposed amendments.

8. SUMMARY

Following on the above trial in the first semester of the year 2000, flexible ongoing assessment was also used for Systems Overview (SO100). The lecturer reported equally encouraging results. The success experienced with this innovative approach to assessing learning will be continued through to programming courses during the coming semester.

APPENDIX A: EXTRACT FROM COURSE OUTLINE

Assessment Plan :

An ongoing flexible assessment scheme is used. Students receive the option of sitting up to six tests, each one worth 10 marks (or 10% of the total course mark), and completing 12 exercises, each one again worth 10 marks (or 10% of the total course mark). If a student completes more than the required minimum of ten items, the best ten results are added to produce the overall total mark. An overall total of at least 80% is required to pass the course.

Test schedule :

Test	Lesson	Topics
1	3	Program development life cycle
2	4	Problem-solving steps
3	5	Problem-solving barriers, breakdown process
4	6	Top-down approach, program design
5	15	Programming languages - general issues
6	15	Programming languages - instructions/syntax

Exercise schedule :

Asgt	Lesson	... due	Topics
1	6	7	Simple algorithm
2	7	8	Algorithm design
3	8	9	Problem analysis
4	9	10	Decision table
5	10	11	Decision tree
6	10	11	Nassi-Schneiderman diagram
7	11	12	Program specifications - A
8	11	12	Simple structure diagram
9	12	13	Program specifications - B
10	12	13	Structure diagram - A
11	13	14	Structure diagram - B
12	13	14	Testing structure diagrams

Arbitration:

If a student is unclear about the application of the assessment plan initial recourse should be taken with the lecturer. It is hoped that a mutual common-sense approach can be taken in applying the plan. In cases of dispute an appeal can be made to the programme leader.

APPENDIX B: SUMMARY OF STUDENT FEEDBACK

Likes

Less stress/pressure than with one [or two] big tests (7*).

Easier to learn [in small bits] (5*); easier on the mind (3*); review work while it is fresh in your mind (6*); revision is easier (4*).

Easier to recognise/evaluate progress (3*); if you do badly in one item it's not too hard to catch up.

Avoids large test at end (3*).

Lots of chances to pass (4*); very lenient.

Encourages/motivates attendance (3*).

Makes you think and understand what is required (2*); makes you think more.

Workload more manageable [because it is distributed] (4*); learning in small sections; less daunting because work is spread out; no need to revise whole syllabus at one time.

Stimulates interest in the course; makes you able to understand the lessons better (2*).

Encourages you to keep on trying for better marks.

Many exercises help retain more information; helps you remember what you learn (3*).

Allows you to focus on the lesson that was just taught; gives subject a chance to sink in;

stuff more likely to "stick in my brain".

Good to apply learning straight away (6*); allows you to practise what you learn.

Opportunity to test understanding immediately after learning; ongoing review; quick feedback.

Requires constant level of work to pass - each session requiring work to be done; makes us work on all aspects of the course.

An otherwise technically difficult subject was made a little simpler.

Brilliant method (2*); great; good to see such dedication from a lecturer; by far the best method employed by any tutor.

Dislikes

Too much work (2*) - possibly, more work required for exercises [surely a plus, OH]; two exercises or tests on same day (3*).

Takes a lot of time; two exercises and only two days time to do them.

Exercises were difficult, not always easy to understand; questions can be misleading.

Passed early, so lost motivation to do later exercises; motivation flagged after reaching an 85% score; easy to slack off at the end if you've already passed; some students passed earlier and left.

Hard to translate into real life sometimes.

Comparatively bad for students who do this course with another tutor (e g in evening).

Hard to get merit - too many items [gives you more chances for merit, OH]; someone can pass all the tests and exercises but not get 95% so not get a merit.

Can't learn from test; too little content in test (small mistakes cost a whole mark).

Sometimes other students would copy each other's work - unfair; too easy to cheat and copy off others; help from other students or those who have already taken the course.

Harsh marking.

Late work not accepted [so many exercises to choose from, OH].

Hard to tell when exercises are due in [date on the exercise handout, OH].

Does not occur for DT100 [hardly a “dislike”! OH].

Some class mates always joke in class.

Dealing with loops and arrays make me confused - but if I go to class each day I know I will succeed. [This student did not go to class each day - and did not succeed!]

Improvements

Only one exercise per session.	Noted, will do.
Slightly fewer exercises (2*).	Yes.
Only one exercise per week.	Per lesson yes, else no.
Motivate the attendance rate at the end.	Yes, will do.
Vary the exercises handed out [to prevent copying]. (2*)	More work but good idea.
More than one night to do exercises.	Timetable problem.
More even spread of tests and exercises. (3*)	Depends on nature of topic.
More time at end of class to review new exercises.	Maybe, many would leave.
Longer tests with more content.	
Less ambiguous wording for some of test questions; write questions more simply.	
More unlikely that students would have to do resits for DT100 if same method used there. (2*)	
One tutorial hour per week for problem solving.	Maybe Fri at 12.30 if no test?
Make it easier to help students pass.	Already so easy.
Easier questions for people needing to catch up.	*Is this a joke, OH?*
More generous marking.	Generous enough, lol.
Get exercises handed in at 8 am so they can be marked by lesson time.	Wow! No way!!
Accept late exercises.	Can’t do, answers available.
Some way to compensate for students who miss classes.	See proposed amendments.