CS-110/Coursework 2: Set 21st November, due 8th December

Neal Harman

Task

Your task is to write a (very simple:-) student record system. You should be able to store details about students - their name, subject and a student number; as well as the level they have most recently *completed* ranging from 0 (meaning they are still in the first year) through to 3 (meaning they have graduated). It should also store their *result* for each level they have completed as a number 0-100. Your program should allow the user to enter student data for as many students as they want. When they finish entering data, it should print out in a 'nice' way the data of all the students entered so far, including their *average* mark over all the levels they have completed.

Strategy and Advice

1. The point of this coursework is for you to start working with *classes* and *objects.* So you **must** use a class to represent the student data.
2. Your first task is to define a *Student* class - the way to do this is to think of all the data that the class must contain (hint: it's all listed in the first paragraph:-) and then to work out what methods you need to *set* and *access* that data; as well as *calculate* any additional information you need.
3. Think carefully about the *Student* class. The methods you define for it are going to be the *only* way you can access and manipulate the students' data (if you define them properly). So make sure you think about them carefully.
4. Once you have worked out what methods you need for the student class, you need to actually write them. There are going to be quite a few (it will depend exactly on how you approach the problem) but they should all be very short.
5. Finally, you need to define *another* class – the one that will contain the *main* method - that reads student data from the user, creates new *Student* objects, and stores them somehow. You *could* store them in an array – but remember that the task says that the user can enter as many students as they want. So you might want to consider using an *ArrayList* instead.
6. One of the slightly tricker things to handle is how to access the data for the students' results for each level. The 'obvious' way is to have a method returning the data for each level - but this would be a very bad way to do it:-)
7. Another slightly tricky thing – also to do with results – is that, depending on what level a student is in, they may not actually *have* results for every level. So you need some way to check *before* you try to access data that does not exist.
8. (In fact, it would be good practice in all cases to try to ensure that if any parts of the data for a student are missing, your program does not fail.)
9. While I don't want to encourage anyone to leave coursework to the last minute, I will – as before – be going over material relevant to this coursework over the next two weeks.
10. Come and ask me questions and discuss it in tutorials if you have them.
11. If you want to extend the problem and do more you can *but in this case there are no specific marks for that.*

How Will This Be Marked

The marking criteria for this assessment is as follows.

1. A Working Solution – regardless of style, a working solution will get 40% of the marks. Solutions that do not work will get less, depending on what and how much is wrong. Solutions that 'work' but don't include an explicit *Student* class to actually contain and manage the students' data will get ***no*** marks for a working solution.
2. Choice of Methods for *Student* – thinking carefully about what methods you need for the *Student* class is very much worth the effort as 30% of the marks will be for this.
3. Style – regardless of whether or not it's working, 30% of marks will be available for style. That is, layout and indenting, commenting, sensible names for identifiers etc, and the layout of the program's output.

Submission

You must submit this work by 11.00 on December 8th 2011 via Blackboard. The standard penalties for late submissions will be applied: 10 marks per day for the first seven days and zero thereafter. If you submit late and believe you have extenuating circumstances then you need to contact me as soon as you can. However, no marks will be awarded to work submitted after the solutions have been released in any circumstances. Marked work with feedback will be returned at most two weeks after submission.