CSC1021 Programming I

Project 1

The purpose of this project is:

- Demonstrate that you can use the Java features covered in the lectures up to and including Week 5, for instance
  - Use of classes and methods
  - Use of selection and repetition statements
  - Use of arrays
- Demonstrate that you appreciate the software engineering aspects of
  - Following style rules in your programs
  - Testing that your program produces correct results
  - Producing project documentation

Program Specification

The School of Computer Science (SOCS) has commissioned you to write some software to keep a track of Stage 1 student marks. Students take 6 modules in Stage 1. Each module has two components - Exam and Coursework and each module can have a different mark for each. The returned mark for a module is calculated as follows:

computed module mark = ((coursework mark * coursework weighting) + (examination mark * (100 - coursework weighting))) / 100

If the exam mark and coursework mark are greater than or equal to 35 then the returned mark is the computed module mark

However, if either (or both) the exam or coursework mark is less than 35 then the returned mark is the minimum(35, computed module mark).

All marks are rounded to the nearest whole number.

The student's performance on a given module can then be recorded as one of

- **Pass**, if the module mark is at least 40
- **Compensatable Fail**, if the module mark is less than 40 but at least 35
- **Fail**, otherwise

Once all the module marks have been determined, a Stage average may be computed. This is found by averaging the returned module marks. The Stage result can then be computed as

- **Pass**, if all modules are recorded as a Pass
- **Pass By Compensation**, if the Stage average is at least 40, no module is recorded as a Fail, and there are one or two modules totalling at most 40 credits recorded as a Compensatable Fail.
- **Fail,** otherwise

Note that this is similar but not identical to how marks are calculated in reality but issues to do with core modules have been ignored

There are two parts to the work that SOCS requires.

Part 1

SOCS want you to produce a `MarkCalculator` class with two public methods:

1. A method `computeMarks` that, given an array of student exam and courswork marks, returns an array of returned module marks.
2. A method `computeResult` that, given an array of student exam and courswork marks, returns a Stage Result for that student.

Part 2

SOCS want you to produce a thin-bar chart, showing a range of bars for Each module . The total height of each bar should be proportional to the returned mark

SOCS suggest that you

1. Take a copy of the `Bar class`. This is similar to the `Square` class but allows you to draw rectangles (bars) by specifying a width and a height with the `changeSize` method. In particular you can draw thin lines by specifying a small width. Note that you will also need the `Canvas` and `Colour` classes used in the Coursework Exercises. *Do **not** make any changes to these classes.*
2. Produce a `StudentChart` class with a constructor with one parameter, an array of integers specifying exam and coursework marks. Provide a `draw` method that draws thin bars based on the values in that array using the `MarkCalculator` class you produced in Part 1. The chart should be annotated with x- and y- axes (labelling of axes is **not** required). Failing marks should be coloured red, Compensating marks yellow, Pass marks green and First class marks (70 or over) magenta. The axes should be black.
3. Provide also a `printSummary` method that neatly prints a table of returned marks corresponding to your chart.

4. Produce a `Summary` class with a `studentSummary` method that allows the user to input exam and coursework marks and then uses a `StudentChart` object to produce a thin bar chart and the corresponding table.

What To Submit

For this project you must provide documentation in a single MS Word file consisting of the following sections:

1. **Header Sheet:** identifying the author, the module, and which piece of coursework this is.
2. **Testing:** Show how you tested the program and discuss why you know that it works correctly.

- **Code Listings:** For this project it is assumed that you will submit the code from three files containing class definitions, one for each of `MarkCalculator`, `StudentChart`, and `Summary`. All of your code should be suitably commented. You don't need to provide listings of code for classes that have been provided for you.

You are reminded that the markers for this project will be looking to see that your code adheres to the style guides that have been described in the lectures. They will also be wanting to see correct use of the control-flow abstractions of Java, and the sensible use of methods, parameters and constants.

Your Java code will contribute 80% of the marks for this project, with the remaining 20% coming from Program Documentation.

Deadline

This Program Documentation and the Java source files **must be submitted to NESS by 16.00 on Friday 13th November**. NESS will accept up to 10 files in case you have chosen to implement other classes in your solution to this problem. Remember, you do not need to submit code for classes that have been provided for you.