COMP101 Lab 8: Fifth Assessed Coursework
Late Penalties

worth 16% of the final mark
assignment five of seven

Failure to submit this assignment or the submission of work that is deemed not to be a reasonable attempt will result in the failure of the module. The completion of the implementation and report as described below will obtain 90% of the marks. To obtain the final 10% students should also complete the Extended Requirements. This may involve doing some additional reading beyond what has been presented in lectures or more complex programming or algorithms. Only attempt the extended requirements if you are confident with programming.

Learning Outcomes. This addresses the following learning outcomes, to

- be able to implement, compile, test and run Java programmes, comprising more than one class, to address a particular software problem;
- understand how to include arithmetic operators and constants in a Java program;
- demonstrate the ability to employ various types of selection constructs in a Java program.
- demonstrate the ability to employ repetition constructs in a Java program.

Requirements. Design, implement and test a Java program that calculates and displays information about penalties for late submission for a piece of coursework as described below. The piece of coursework is given an original mark which is an integer between 0 and 100. The passmark for the piece of coursework is 40%. Each piece of coursework may be subject to a late penalty if it is submitted after the deadline. There are two ways of calculating late penalties.

Scheme 1 The work loses 5 marks every day it is late. If the mark drops below 20, no further late deductions are made and the mark stays as 20. If the original mark is initially less than or equal to 20, then the mark stays as its original value.

Scheme 2 The mark for the work is reduced by 10% of the current mark every day it is late. If the mark drops below 25, no further late deductions are made and the mark stays as 25. If the original mark is initially less than or equal to 25, the mark stays as its original value.

You should write a program in Java to input a mark (m) and a number of days (d). The mark (m) should be an integer between 0 and 100 (inclusive). The number of days (d) should be an integer between 0 and 20 (inclusive). If the user inputs a value outside these ranges (for

the mark and/or number of days), this should be disallowed and the user asked to re-input the value until the value is within the correct range. For each scheme print out the resulting mark for each day between 0 and d days (inclusive). For the second scheme the marks should be shown as real numbers (rather than integers). Also, for each scheme report how many days late the assignment can be before it is given a failing mark (note that this number of days until failure might be larger than the number (d) of days that the user input to display). You are expected to use Java's loop constructs to achieve this.

Hints

1. Write your program with self-contained classes and methods.
2. The input values m and d should be checked to ensure that they are in the specified ranges.
3. For scheme 2, don't worry about rounding your values but you should print them out to two decimal places (using printf or otherwise).

Examples

```
Please input mark: 82
Please input number of days to display: 10
Scheme 1
(0) 82  (1) 77  (2) 72  (3) 67  (4) 62  (5) 57  (6) 52  (7) 47  (8) 42  (9) 37  (10) 32
This work can be up to 8 days late before failing.

Scheme 2
(0) 82.0  (1) 73.80  (2) 66.42  (3) 59.78  (4) 53.80  (5) 48.42  (6) 43.58  (7) 39.22  (8) 35.30
(9) 31.77  (10) 28.59
This work can be up to 6 days late before failing.
```

Extended Requirements. Rewrite your programs to make them more general, allowing input for *all* parameters. Namely, you should allow the input of the mark (m) and number of days to display (d) as previously. However, you should also allow the user to input all the other parameters namely, the pass mark (previously 40), the limit below which no further reductions are made (previously 20 for scheme 1 and 25 for scheme 2), and the number of marks lost per day for being late (previously 5 marks) *or* the percentage reduction per day (previously 10%). Each of these values should be whole numbers between 0 and 100 inclusive. You should check the input values are in the correct range and that only one of the percentage reduction or number of marks lost per day is input. Output the resulting marks for the number of days (d), given the input data for the resulting scheme. There should be just one display of days late/marks (not two, one for each scheme as previously).

Submission Instructions. You must submit all of your files (PDF report and Java source files) as a single compressed zip file. Do not use any compression method other than zip. (Note that this is the compression method that the Windows and Mac file managers use to compress files.)

Your submission, should consist of a report (a PDF file) and implementation files.

- The report (a PDF file) should consist of

  Requirements: Summary of the above requirements statement.

  Analysis and Design: A short (one paragraph) description of your analysis of the problem including a Class Diagram outlining the class structure for your proposed solution, and pseudocode for key methods. Note that your solution should comprise (at least) two classes: an "application class" and a "target class".

  Important: Make certain that your main application class (the class with the "main" method) is called "LatePenaltiesUser".

  If you do the extended requirements, make certain that your main application class is called "LatePenaltiesUserExt".

  Testing: A set of proposed test cases presented in tabular format including expected output, and evidence of the results of testing (the simplest way of doing this is to cut and paste the result of running your test cases into your report).

- The implementation should consist of

  Your Java source files, i.e. the relevant .java files, not the class (.class) files.

Upload your files (as a single compressed zip file) to

https://sam.csc.liv.ac.uk/COMP/Submissions.pl

(you will need to log in using your Computer Science username and password).

Submission Deadline. Monday 28th November, 5pm.

Mark Scheme Marks will be awarded for

- Analysis and Design 15%
- Implementation 60%
- Testing 15%
- Extended requirements 10%

Please see the module web page for the feedback form.

Note.

- Because submission is handled electronically, ANY FILE submitted past the deadline will be considered at least ONE DAY late. Late submissions are subject to the University Policy (see www.csc.liv.ac.uk/department/regulations/practical.html).
- Please make sure your Java classes successfully compile and run on ANY departmental computer system, both Windows and Unix. For this reason the use of powerful IDEs like NetBeans is discouraged.
- Note this is an individual piece of work. Please note the University Guidelines on Academic Integrity (see https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf).