CS2035 - Assignment 1 - 2019

Out: January 8th, 2017

In: Sunday, January 27th, 2019, at 11:55pm via Owl

Introduction

You are required to write a program in a single file called **ass1_2017.m** that does the grade calculation for CS2035. Normally the professor does this but he in incredibly lazy and so has out-sourced the job to you as assignment 1.

You are suppled with a file `ass1_template_2019.m` that contains much of the code you need to complete this assignment. Write the additional code needed to complete the commented tasks to get a working program. The tasks to complete this assignment are enumerated below:

1. Write the `fprintf` statement to print all the read data for each student. This statement should use appropriate formats to print character strings and numbers. For example, formats %s and %d (in their various forms) are needed.

2. Compute the `lab_total` variable for student `i` as the sum of all the labs, up to a maximum of 10

3. Compute the assignment grade as 9% for assignments 1 to 3 (multiple the assignment grades by 0.09) and 8% for assignment 4 (multiple the assignment grade by 0.08).

4. Compute the exam grade as:

   (a) If the student's midterm mark is greater than his/her final exam grade, compute the `exam_total` variable as 0.2 of the midterm mark and 0.35 of the final mark.

   (b) If the final mark is greater than the midterm mark for the student then compute the final exam grade as 0.55 of the final mark.

5. Set the message character string for rule 1 (the 45% rule). Both the assignments grade and the exam grade must be 45% of the maximum values. If this is not satisfied, the course grade cannot be $\geq$ 45%. Keep track of the number of times rule 1 is invoked for the fprintf statement at the end of the program (use variable `rule1_no`).

6. Set the message character string for rule 2 (the 60% rule). Both the assignments grade and the exam grade must be 50% of the maximum values. If this is not satisfied, the course grade cannot be $\geq$ 60%. Keep track of the number of times rule 2 is invoked for the fprintf statement at the end of the program (use variable `rule2_no`).

7. Set total and count variables for average calculations The `passed_course_total`, `failed_course_total` and `passed_course_ct` and `failed_course_ct` variables are set below this comment as an example of what you have to do.

8. Compute the counts for grades $\geq$ 90, $\geq$ 80, $\geq$ 75, $\geq$ 70, $\geq$ 65, $\geq$ 60, $\geq$ 55, $\geq$ 50, in the range 1 and 49 and equal to 0 - this code requires a nested if-then-else statement,

Your program must be documented with good grammatical correct comments and your program should be nicely indented. The course webpage has examples of 2 well documented MatLab function examples, **commented_new_decimal2roman.m** and **commented_new_roman2decimal.m**. Use a block comment at the beginning of your program in ass1_2018.m that gives your full name (as it appears on your student card), your student number, your uwo email address and which assignment you are doing. For example:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  John Barron                      %
%  123456789, barron@uwo.ca         %
%  CS2035, Assignment 1, 2019       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

would be my block comment. Also include a brief comment about what your program does (not done above).

The source code **ass1_2019.m** should be submitted via OWL by the due date. You know how to submit your assignment because your did the first lab! We will run your code while grading your assignment. It is necessary to follow the assignment specs to optimize your grade.