

COMP 6411: ASSIGNMENT 1

DUE DATE: May 20, 11:55 PM

Note that assignments must be submitted on time in order to received full value. Appropriate late penalties (< 1 hour = 10%, < 24 hours = 25%) will be applied, as appropriate.



DESCRIPTION: In this assignment, you will have a chance to gain experience with the Python programming language. Python is an easy to use, dynamically type Object Oriented language (though you do not have to use objects). It is syntactically similar to C-style languages like C++, C# and Java, but is somewhat simpler to understand.

Your job in this assignment will be to model a task that I undertake at the end of each course. Specifically, you will calculate and display grades for the course. In my case, I use a spreadsheet to do this. In your case, you are going to provide this functionality yourself. The assignment will work as follows.

1] A class list will be provided in a text file called "class.txt". The format will look like this:

```
2345|Joe|Smith
4567|Sue|Jones
9798|Ahmad|Taleb
```

In other words, each record has three fields, separate by a bar symbol ("|"). The fields are <student ID, First Name, Last Name>. Note that each record will have the proper data, so no additional error checking is required to read the contents.

2] Each graded component will be stored in a separate text file. The files will be called:

```
a1.txt
a2.txt
project.txt
test1.txt
test2.txt
```

Each file will be formatted the same way. An example might be:

```
35
2345|26
9798|17
```

The value on the first line indicates the maximum point value for this component. The remaining lines provide an <ID|grade> combination. So, in this case, student 2345 received 26/35 on this component.

Note again that no special error checking is required. The grades will always be in the proper range (i.e., $0 \leq \text{grade} \leq \text{max}$). Moreover, any student ID will correspond to an ID in the class list. However, not all students in the list will have a grade, since some may not have submitted the assignment/project, or may simply have dropped the course at some point.

3] A simple menu (described below), will allow users of your program to list the grades for any component of the course.

4] So that's the easy part. Now it's time to provide more useful functionality. Specifically, you need to provide a very simple spreadsheet-style display. Here, you will combine each of the basic grades with the following two values: The overall numeric total for each student, and the associated letter grade.

5] The numeric total will be the sum of each of the graded components (2 assignments, project, and both tests). This will be a score out of a maximum of 100. To do this calculation, you must *normalize* each grade so that it is mapped into the grade representation in the syllabus. In this case, the breakdown will be as follows:

```
a1: 7.5%
a2: 7.5%
project: 25%
t1: 30%
t2: 30%
```

So, for example, let's assume that the project was actually graded out of a maximum value of 40 points and that a particular student received 32/40. Because the project is worth 25% of the overall grade, this component would contribute 20 points towards the final numeric grade for this student.

6] The associated letter grade also has to be calculated. We will use a very simple mapping for this. We will assume the following possible grades:

```
A+
A
A-
B+
B
B-
C
F
```

To calculate the grade we will do the following. As a default, we will set the Pass/Fail grade at 50. All grades above this point will be mapped into equivalent ranges. Since there are 7 grades above an F, then each grade range will be approximately 7 points. (e.g., a 94 receives an A+, while a 60 receives a B-.

7] Now, of course, we have to display these values. This will be a purely text-based display and will look like this:

ID	LN	FN	A1	A2	PR	T1	T2	GR	FL
12344	Smith	Joe	21	22	43	37	31	88	A
54316	Fish	Wu	27		27	31	26	62	B-

Note the following:

- The columns must be lined up evenly in order to be able to read the spreadsheet. To make things easier, student IDs will always have 5 digits, and first and last names will have a maximum of 6 characters each.
- All relevant columns are listed in the spreadsheet (ID = student ID, LN = last name, FN = first name, A1 = assignment 1, A2 = assignment 2, PR = project, T1 = test 1, T2 = test 2, GR = numeric total, and FL = final letter grade)
- By default, the records must be listed in ascending order by student ID.
- If a student does not have a grade for a given component, the column will be blank (e.g., A2 for Wu Fish). This will be counted as 0 for the purpose of the final grade calculation.

8] So now we have to provide a menu in order to use the program. When the program is first run it will read all relevant text files and then display the following menu:

```
1> Display individual component
2> Display component average
3> Display Standard Report
4> Sort by alternate column
5> Change Pass/Fail point
6> Exit
```

So let's look at each of these in turn:

Option 1: you simply display the relevant component grades. For this to work, you must prompt the user for the component name (A1, A2, PR, T1, or T2). These labels are case insensitive so the user can, for example, enter A1 or a1. Invalid entries should result in an error message and a new prompt. Once the user enters a correct label, the contents would be displayed as follows:

```
A1 grades (35)
1234      Smith, Bob      34
3454      Dill, Mona      23
```

The first line lists the component name and the max value. The remaining lines list the info for each student. Note that the <last name, first name> are listed as a single entry separated by a comma. The values should again be listed in evenly spaced columns for readability.

Once the list is displayed, you will return to the main menu.

Option 2: Again, you will prompt the user for the relevant component and then simply list the average as follows, before returning to the main menu:

```
A1 average: 27.35/35
```

Option 3: Display the standard report as listed above, then return to the menu.

Option 4: There are two optional sort orders: LT (last name) and GR (numeric grade). Here, you will prompt the user to provide one of these two labels, then you will show the standard report, but re-sorted according to the specified sort criteria. The main menu will then be displayed (note that the standard report will again be listed by Student ID, if Option 3 is selected again in the future).

Option 5: In this case, we will simply change the Pass/Fail point so that we can see how final letter grades might be affected. The user will be prompted for the new P/F point, and then the standard report will be listed, this time with the new letter grades. If the report is shown in the future, it will again use the standard default of 50. Once the report is listed, the main menu will be displayed again.

Option 6: Simply print “Good Bye” and terminate the application.

So that’s the basic idea. If you haven’t used Python before, you will find that it is a very accessible language with a lot of documentation and supporting materials online. If you like to code, this assignment may actually be fun.

DELIVERABLES: Your submission will have just two source files. The “main” source file will be called `grades.py` and will contain the logic to read the data files and display the main menu. In

other words, you can run your python program from the command line with “python grades.py”. The second file will be called `compute.py` and will provide all of the functionality associated with the menu options (i.e., calculations and spreadsheet display). That’s it. These are the only two source files that you will need.

Once you are ready to submit, place the two source files into a zip file. The name of the zip file will consist of "a1" + last name + first name + student ID + ".zip", using the underscore character "_" as the separator. For example, if your name is John Smith and your ID is "123456", then your zip file would be combined into a file called a1_Smith_John_123456.zip". The final zip file will be submitted through the course web site on Moodle. You simply upload the file using the link on the assignment web page. Note that you do not submit the data files. These graders will have their own files for testing purposes.

Good Luck