420-LCU-05 Programming in Python - Assignment 3

Due April 17th, 2018 at 11:55 p.m.

1- **Identification section**: The grader and I need this section for the accurate processing of your assignment. If missing, you may lose up to 5% of the mark.

```
""""
Your Name and ID
420-LCU Computer Programming, Section #
S. Hilal, instructor
Assignment 3
""""
```

2- **Submission**: Submit in 1 Python file, with the extension .py. No need to create a ZIP file. Be sure to respect other instructions specified in the assignment. An important part of each assignment is to correctly follow the instructions as closely as possible.

3- **Late assignments** are accepted up to 1 week from deadline. But late penalty will be applied.

For this assignment, you will develop an application that is very similar to Assignment 2 but will be using some of the different concepts and data structures that we have been learning.

Learning Objectives:

- Defining and using a simple class. Class attributes, instance attributes, and methods
- Understanding the concepts of class and instance(s) of a class.
- Using objects to store information.
- Using a list for temporary manipulation of student information
- Using files as an alternative to user input and for permanent storage of program output.
- Using structured formatting to print information

The program will enable a teacher to **input**, **analyze** and **report** on the grades of the students in one class based on their marks in different components of the course. It can calculate a student's **total grade**, **letter grade**, **class average, grade distribution, and give a structured visualization of the stored data as requested**. Your program will be using the **Student** class that we have created in Lab-8 with some further updates to add required attributes and methods. **You may also add other functions to your main program outside of the Student class**.

Please note that you can reuse most of your menu handling code from A-2. **However, some of this code will not be needed as you move it in the Student class definition**. Make sure to remove unnecessary code.

It is strongly advisable to use the basic structure of your program in assignment 2 to build this program. You need to determine what to keep and what needs to be removed, slightly changed or replaced. **As a matter of fact, you will notice that this assignment is much easier than A-2.**

You need to start working on your program early and seek help as needed in order to complete your program by the **deadline which will be strictly enforced** as we reach end of the semester.

1- **Program Input and Description of Data:**

A student can be identified by a name (first name only) and ID (3-digits). For each student, there are 6 grades based on 2 tests (/20 each) and 4 assignments (/15 each). The grades are **integers**. The program should be able to process input for any number of students but you can limit your tests to the list of students as provided in the Input file **students.txt**

The input to your program (students' data) will be read from a text file (students.txt). **Note that the data in the file is of the same format as the user input in A-2**; Separated by commas and no spaces. Each record is on a separate line.
For each student, your class must have code to **calculate and store the total grade and corresponding letter grade when an instance is created and also following any grade update**.

Note that letter grade calculations are same as per A-2

2- **Storing students' data**:

```
students_list = []   # List of all student instances
```

Each student record is created as an instance of the Student class. **students_list** will be used as your storage manager to keep track of all the Student instances. The list will give the entry point to each student instance and will be used to access the data for all the required tasks. Each list item provides access to the record of one student.

Running Your Program:

The program starts by giving the user a menu of options to select from. When the program completes the processing of a given option, the menu will be displayed again for another selection and continues until the user selects option 6, Exit.

Sample Run:

Welcome to the Teacher's Simple Class Calculator. Here's the list of options:

1- Read and process all students' records
2- Display All student records including total and letter grades and class average

3- Display the complete record of a particular student
4- Update a student's grade
5- Display a list of all students who achieved a particular letter grade.
6- Exit

Select an option by entering its number or 6 to exit:

Description of the Different Options (above):

1- **Option 1**: Read the text file provided and store student data. Calculate and store total grade and letter grade for each student. You can assume that all records are valid and that there are no duplicates. Duplicate names may exist but no duplicate IDs.
2- **Option 2**: Display the information stored for all students in a table format. Class Average.
3- **Option 3**: The program will ask the user to enter the name and ID of the student. The program will print the student's record. Same format as in option 2. If the **name and ID** do not match a stored record, an error message is displayed.
4- **Option 4**: Similar to Lab-8, update a particular grade for a particular student.
5- **Option 5**: Display the requested list. Same format as in option 2.

The program will display the main menu following the completion of each option and until the user selects option 6.

Please add error checking as you see required. Add comments.

Add docstrings to all methods and functions.