420-LCU-05 Programming in Python - Assignment 2
Due April 06, 2018

1- **Identification section**: This section must be either in a comment, with a '#' preceding each line, or enclosed within triple quotes ('''). The grader and I need this section for the accurate processing of your assignment. Assignments missing this may lose up to 5% of the mark.

```
"""
Your Name and ID
420-LCU Computer Programming, Section #
Friday, April 06th
S. Hilal, instructor
Assignment 1
"""
```

2- **Submission**: Submit your assignment 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 a small application that will enable a teacher to enter, **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**, **rank** within the group and the **class average**. Your program will define some useful functions that can be used to do some of the required operations.

Learning Objectives:

- Practice using lists, strings and loops
- built-in functions and methods for strings and lists
- User-defined functions.

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 can be integers or floats. The program should be able to process input for any number of students but you can limit your tests to 10 students max.

Your program will define the following lists that can be used to store the students' information:
Students, IDs, T1, T2, A1, A2, A3, A4 = [], [], [], [], [], [], [], []
You can create any other lists or variables that you may need.

Below is a table of the letter grades that correspond to the total score:

Total Grade	**Letter Grade**
87 or above	A
From 75 to 86 inclusive	B
From 65 to 74 inclusive	C
Below 65	F

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 give option, the menu will be displayed again and user prompted to make another selection and continues until the user selects exit.

Your program must do the required tests to make sure that it does the valid options only. (E.g. cannot process option 2 if students list is empty). Define other tests and add in your comments.

Sample Run:

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

1- Enter student records (Name, ID, and 6 marks separated by commas)
2- Give the total grade, letter grade, and rank of a student
3- Calculate the class average
4- Display a simple bar chart to show grade distribution.
5- Exit

Select an option by entering its number or 5 to exit:

Description of the Different Options (above):

1- **Option 1**: The program will keep asking the user to enter a student record until the user enters "done". **Computer output in bold**. User input shown in red
Example:
Enter Student Record (Separate by commas, no spaces) or done: Anne,234,20,18,12,10,15,11
Record Accepted
Enter Student Record (Separate by commas, no spaces) or done: Bob,124,15,18,12,15,15,12
Record Accepted
Enter Student Record (Separate by commas, no spaces) or done: Greg,124,15,18,12,15

```
Record Incomplete
Enter Student Record (Separate by commas, no spaces) or done: Greg,124,15,18,12,15,13,12
Record Accepted
Enter Student Record (Separate by commas, no spaces) or done: Bob,124,20,18,12,15,15,12
Record already exists. Do you want to updated record (yes/no): yes
Record Updated
Enter Student Record (Separate by commas, no spaces) or done: done
```

- If a student record has already been entered, the program informs the user.
- There may be duplicate names but IDs are unique.
- The program rejects an incomplete record.
- The program will then store a complete record in the defined lists.

2- **Option 2**: The program will ask the user to enter the name and ID of the student. The program will print the numeric grade, letter grade and rank.

```
Enter the name and ID of the student: Anne,234
Grade for Anne ID = 234: 86 B Rank= 2
```

3- **Option 3**: The class for the entered grades will be displayed.

```
Class Average = 86
```

4- **Option 4**: Graph will be displayed.

Writing your program:

1- Your program must define at least 2 functions to help in doing some of the calculations and analysis. Remember to make your program as modular as possible (the more functions you develop the better)

2- Here are some of the possible functions:
   - Read and store a student's record
   - Calculate a student's raw score, letter grade and rank.
   - Check if a student's record already exists

3- You can use any of the list/strings built-in functions and methods that we have seen in class.

4- The program will display the main menu following the completion of each option and until the user selects option 5.

Testing Your Program:

The program does not store any data between runs. It is a good idea that you create a set or 2 of data that you can use to test your program and always use the same. You can store your data in a comment at the beginning of your program. The more records you use to test the better.

```
# Anne,234,20,18,12,10,15,11
# Bob,124,15,18,12,15,15,12
# Greg,124,15,18,12,15,13,12
```