CS 111 – Introduction to Computer Science – Fall 2017

Lab Assignment #10

File Processing * *(30pts)*

Due Date: at 11:59pm on Saturday, Apr 28.

In this lab, you are going to write functions that process data stored in files.

Before getting started with the lab, copy the entire `lab10` folder from the course folder (`H:\Compsci\givens\cs111`) to your `U:\cs111` folder.

Grade Distribution

You are to write several functions that read sequences numerical grades and compute grade information across the common letter grades (A: 90-100, B: 80-<90, C: 70-<80, D: 60-<70, and F:<60).

You are to design and write a program (`gradeProcessing.py`) with the following functions

- `gradeDistribution(inFileName)`
  Receives the name of the input file, opens it, reads the grades in the file one line at a time, and outputs a grade distribution and histogram (to the console) based on the grades in the file. A grade distribution is simply the percentage of grades that are classified as each letter grade. For example, if the file contains 2 A's, 4 B's, 3 C's, 2 D's and 1 F, then 16.6% of the grades were A's, 33.3% were B's, 25.0% were C's and so on. The function would produce the following output for the grade distribution (percentages written to one decimal place):

```
Grade Distribution:
   A: 16.6%
   B: 33.3%
   C: 25.0%
   D: 16.6%
   F:  8.3%
```

  A histogram is a visual representation of data. For this histogram use `*`s, where each `*` represents one percentage point. For example, the previous data would produce the following histogram

```
Grade Histogram:
  A: ****************
  B: **********************************
  C: ************************
  D: ****************
  F: ********
```

*Based on the labs of Dr. Rance Necaise

Have a single blank line written between the Grade Distribution and the Grade Histogram.

Test your function using a separate driver file (import the functions from `gradeProcessing.py`). Use `grades1.txt` and `grades2.txt` as input files to test your function, and print percentages to one decimal place.

- `classGrades(inCSVname,outCSVname)`
  This function receives the name of an input file and an output file, both CSV files, and reads grade information for several students and outputs (to the output file) final grade information for each of the students. Each line of the input file contains a student name followed by multiple comma delineated grades. The function averages the grades for each student and outputs (to the given CSV output file) the student name, grade average (to one decimal place), and letter grade.

  For instance, if the input file contained the following information:

  ```
  Sally,75,77,83
  Jane,92,85,90
  Beth,97,86,92
  ```

  Then the output file would contain the following:

  ```
  Sally,78.3,C
  Jane,89.0,B
  Beth,91.7,A
  ```

  Don't forget to open, write, and close the output file appropriately.

  Test your function using a separate driver file (import the functions from `gradeProcessing.py`). Use `class1.csv` and `class2.csv` as input files to test your function, and write percentages to one decimal place. Use output file names `class1-out.csv` and `class2-out.csv`.

Your functions must meet the following requirements.

- Include an appropriate file prolog at the top of your source file.
- Include a comment immediately above each of the two functions that you write to briefly describe the purpose of the function.
- Use meaningful variable names.
- Include appropriate comments throughout the functions.
- The module will not contain a *main* function. Nor should it contain any executable code outside of the functions. To test your functions, you should create a separate driver module that imports each of the functions.

■

Finishing Up

When you are finished with the lab, you need to show me that your code runs and correctly computes the solution for each part of the lab. Also, you need to submit the source files for grading. To submit the files, find the lab assignment on Canvas and upload the two files:

- `gradeProcessing.py`

You do not need to turn in your driver file that tests your functions. Remember, all of the files must be named exactly as indicated above, with the same case and with no spaces or special characters.