CSE 174 Lab 5 – Grade Calculator
Fall 2024
Assigned: 9/27/2024
Due: 9/28/2024

Introduction:

A Biology Professor, Dr. Anastasia Shimitakis, has many text files containing student first names and grades for her classes. She wants the grades automatically transformed into letter grades, along with the number of students in her class counted and their average grade totaled. These letter grades and averages she then wants saved into another file, but this one with no identifying information (i.e., no names). She could buy software to do it herself, but that costs money. Therefore, she's asked you to write the program for her.

She has an example class with fake students and grades inside a text file called **class1.txt** (download the file from the Lab Canvas page) which looks like the following:

```
Jerry     86
Eric      97
Karen     90
Valerie   72
Patricia  99
Jim       62
Han       80
Scott     94
Derek     0
Marilyn   88
Elyse     26
Arthur    89
Winni     88
Jessie    41
Addison   93
Homer     65
Max       17
Carmela   72
Briget    61
Bob       11
Blanca    63
Roseline  95
Kelly     100
```

You can assume all grades are always integer values.

She has also created a mockup of what she wants the application to look like. This application should take in the name of the file to be read and the name of the file where the data can be saved to. Then, after the program processes the file, it will output the count of the number of students and the class average to the screen.

```
Enter an input file name: class1.txt
Enter an output file name: class1_out.txt
Number of students: 23
Class average: 69.09
```

Additionally, she wants to output the results of each row to the file for error checking. Check the example results from her **class1_out.txt** file:

```
1   B
2   A
3   A
4   C
5   A
6   D
7   B
8   A
9   F
10  B
11  F
12  B
13  B
14  F
15  A
16  D
17  F
18  C
19  D
20  F
21  D
22  A
23  A
24  Number of students: 23
25  Class average: 69.09
```

Dr. Shimitakis does not use the + or – modifiers for grades, so you can just use the standard letter grade scale relating to the below table:

Student Score	Letter Grade
Greater than or equal to 90	A
Greater than or equal to 80	B
Greater than or equal to 70	C
Greater than or equal to 60	D
Less than 60	F

How to Start Writing This Program:

The following steps is a thought process or writing this program. You will need to plan for what you need to do and write that down, then start implementing it step by step.

1. Getting an input filename from the user and opening the file.
2. Getting an output filename from the user and opening the file.
3. Start reading from the input file **for each line** inside the file.
4. **strip()** and **split()** the line to remove the hidden newline character and to break it into pieces. Loop through the line **for each item** on the line.
    a. Skip over the name, we do not want any identifying features. Or more specifically, we only want to loop for the number (recall isdigit() exisits).
    b. Determine the letter grade from the score (**convert to an int value**) following the student's name.
    c. Write the result inside the output file.
    d. Add up the original score so at the end (after the loop is done) the summation can be used to calculate a class average score to be displayed in the console as well as be written inside the output file.
    e. Count how many students are read so far from the input file so at the end (after the loop is done) it can be displayed properly.
5. When the loop is done, calculate the average and display the student count and average in the console.
6. Print the student count and average inside the output file.
7. Close the input and output files.

Follow a Plan:

These steps are a simple and effective way of starting a program:

1. Coming up with an idea.
2. Coming up with a strategy for implementing the idea.
3. Writing down everything step by step.
4. Implementing the code step by step.

As a programmer, you need to get used to doing the same thing. As you practice more and you get faster and faster, you will start to plan the strategy in your head without writing it down! But, all programmers started from writing out all the steps. So, create a new Python file in Visual Studio Code, call it **lab_5.py**, and start implementing all the steps.

Test Your Code Comprehensively:

As a programmer, you always need to test your program using different inputs to make sure everything in your code works as expected. Create new text files and test your code with them. Check the result to make sure that your code is generating the exact result. You can always assume the user gives the correct filenames and there are always acceptable positive numbers inside the given input file.

Additional Tests:

There are additional tests on the CODE plugin **that your code needs to pass them** in order to get full points. These additional tests simulate a real world situation when other people use your program and you never know how many values might be used inside the input file.

- One of the questions that you might ask yourself or your professor is:
    - I can't see what values the CODE plugin is using to test my code with, so I can't fix my code when I can't see them!
- However, in the real world you man not actually be able to get this information, hence why we are giving you practice now. It is your responsibility to make sure your code can handle anything we throw at it!
- Think about all possible cases that your code can't handle and **ask your professor about them** to see if you code needs to solve them **instead of asking your professor what the inputs used in the additional tests are**.
    - For example, you can ask:
        - Should I be worried about dealing with negative numbers in the input file?
        - Should I be worried if the input file is empty and there is nothing inside the file?
            - By the way, the answer to those two questions is **no**. You don't need to worry about these two situations.

Submission Instructions:

After you have completed the lab assignment, locate your source code (**lab_5.py**) in your workspace and submit it to the corresponding Lab 5 CODE plugin.

Rubric:

Note: If you submit your assignment and it does not compile, you will receive a **zero**.

Task	Grade
Program produces correct output and passes all test cases.	35
Program successfully reads the data from the file loops.	15
Program successfully outputs to a file the letter grades, number of students, and correctly calculated class average.	15
Followed proper programming practices such as using descriptive variable names, finding a reasonable minimal solution, proper logic, avoided hard coded values, and used loops properly.	20
Program closes the files.	5
Program has descriptive comments at the top of the program and inside the code.	5
The program had no style errors as indicated by the CODE plugin.	5
Total	**100**