CS21 Module 7 Homework

Please use the given name for your scripts!

1. (grade_reports.py). Consider the grading for a fictional course. Every graded assignment has 2 values: the grade for that assignment, and the total number of points possible on the assignment. For instance, on a lab a student completes everything except the 5-minute brainstorm, resulting in a score of 9/10 (a 90%). Over the semester, several assignments are given, each with their own score and a total number of possible points. To find the final grade in the course, you would divide the sum of all the scores for that student by the sum of all the total possible points for each assignment. That percentage can then be used to find the letter grade for the course. (Please note: this is NOT how we calculate grades for this course. It is a simplification for the assignment)

   Consider the example student Ben Linus. In our student_grades.txt file, we can see his name, followed by his grades. Each assignment's grade is broken up into 2 lines. The first line is their score for the assignment, the second is the total number of possible points for that assignment. The comments are added by me for explanation and do not appear in the file. Each student has exactly 6 assignments, and your program can assume this will always be the case.

```
Ben Linus
50 # Score for assignment 1
50 # Total possible points for assignment 1
72 # Score for assignment 2
75 # Total possible points for assignment 2
75 # Score for assignment 3
100 #Total possible points for assignment 3
20
25
23
25
83
100
```

   To find Ben's grade we will use the formula:
   Grade = sum of all scores / sum of all possible points

   In this case, Ben's grade would be:

   (50 + 72 + 75 + 20 + 23 + 83)/ (50 + 75 + 100 + 25 + 25 + 100) = 86.13% which is a B

   We want to write a program that can read in a list of students from a file (`student_grades.txt`) and their grades on different assignments. Then the program will calculate the final grade for their student and find their letter grade. It will then create a file that will list each student and their final letter grade.

Additionally, we also want to know some general statistics about the overall grades in the course. What was the highest overall grade in the course? The lowest? The average overall grade. Additionally, we want to see a histogram showing how many students earned each letter grade in the course. We will produce these statistics and report them in a separate file from the individual letter grades for each student so we can easily share overall course information without sharing any individual letter grades.

If we had 3 A's, 5 B's, 6 C's, 3 D's and 1 F, our histogram would look as follows:

```
A's: ***
B's: *****
C's: ******
D's: ***
F's: *
```

Functions:

Your code must have and use the following functions. All other code should be in the main function.

Function Name	Input	Processing	Output
`get_file_name`	A string variable to specify which file name we are asking for. Examples include "Grade Report", "Grade Input", etc.	Will ask the user for the name of the file, specifying which file we are asking about	Return the file name provided by the user.
`determine_grade`	The numeric grade for the student	Finds the letter grade that corresponds to the numeric grade. 90 and above is an A, 80 and above is a B, etc	Return the letter grade for that students
`histogram`	The category label, the count for that category, the name of a file you want to append to.	Prints the line of the histogram for that category to the output file. If "A's" is passed in as the category with a count of 3: `A's: ***` Will append the line to the output file. The function should open and close the file	Does not return a value, just prints to the file

Exceptions:

There are several opportunities for exceptions to occur in this program, such as opening files, reading in values from the files, working with values read in from the file, etc. You must catch and handle all exceptions. It is not acceptable to put all of your code in one try suite, your try suites should be specific to one part of the program that could have an exception. How you

handle that exception will depend on what you expect to happen. For instance, if you are unable to open the input file, then there is not much the program can do, so it should print an error message and exit. If you can't open the grade report file, then that file cannot be produced but that does not prevent the output file with each student's letter grade from being produced.

Reminders (not following will result in point deductions):

If a function's task doesn't include output to the user, do not do it!

All programs should have a `main`() function

Use constants! No magic numbers! No global variables!

It is expected that you will complete the same process of development that we use in class. When you reach the point of having an algorithm (pseudocode), this will become the comments of your program as a starting point for writing code. Comment first, then code!

Be sure to include comments at the top of the program that include your name, class and a short description of the program.

Each function should begin with a comment describing the task the function will perform.

Be sure all output is formatted. Unless otherwise, specified, displays non-integer values with 2 digits after the decimal point.

Any work you submit for this assignment should be authored entirely by yourself. Assistance is permitted from the instructor or teaching assistants only. All submitted programming assignments are subject to originality verification through software designed and used for the Measure Of Software Similarity (MOSS).

TIPS:

Some of these tasks may remind you of previous assignments. Have we written any of these functions before?

This is a long program for us. Don't wait until you have everything coded to try and test it. Develop your code in smaller steps and test along the way. That will make it easier to locate the cause of the errors. For instance, you could write your functions first, and test them individually before working on the main. When reading in the input, first just have it print what it reads in to the screen to make sure the file input is working (remember to delete these print statements later). Have it find the letter grade for just one student and make sure that works before having it run on all students. Have it produce the letter grade file and test to make sure that is working before adding in the grade report.

We have provided example output files. Check those if you are confused about what to do.

Sample Runs:

```
What is the name of the file to use for grade input? not a file
Could not open input file. Exiting now
```

```
>>>

= RESTART: C:\Users\kaplis\Desktop\OneDrive - University of Vermont\CS
021\Week8\grade_reports.py

What is the name of the file to use for grade input?
student_grades.txt

What is the name of the file to use for outputting individual student
grades? letter_grades.txt

What is the name of the file to use for Grade report? report.txt

>>>
```