Programming Assignment #1

Grade Book

CS 2308.251, 252, and 257 Spring 2016
Instructor: Jill Seaman

Due: Monday, 2/8/2015: upload electronic copy by 9:00am

Problem:

Write a C++ program that will compute the final average and assign a letter grade for the students in a class.

Input: The input file named "class_scores.txt" will have the following format:

one line for each student containing the following, separated by at least one space:

student ID
5 exam scores
1 participation score
4 assignment scores

example:
A00123456 95 88 84 77 0 50 10 7 8 10

Assume the following (you do **not** need to validate these things):
- The student ID will not contain any spaces.
- Each score is an integer.
- There is no missing data.
- The data follows the format described above.
- There will be no more than 50 students (but at least 1).

See the website for a sample input data file.

Note: the input file may have blank lines at the end. Your program should ignore them.

Output:

Write the output to an output file named "final_grades.txt".

The program should output the **student ID**, the **total points** (computed according to the instructions below), the **percent grade** formatted to 1 decimal place and the **letter grade** (computed according to the grading scale below) **for each student**.

The grading scale:

Total Points	Grade
448-500	A
398-447	B
348-397	C
298-347	D
0-287	F

The instructions for computing total points:

Exams are out of 100 points
Participation is out of 60 points
Assignments are out of 10 points

The lowest Exam score is dropped.
The maximum Participation score allowed is 60 points (even if the score in the file is greater than 60).

Here is an example of how the output should appear.

```
StudentID  Points  Percent  Grade
--------------------------------

A00123456    449     89.8     A
A00529154    418     83.6     B
A00656624    420     84.0     B
A02507691    403     80.6     B
A00612352    397     79.4     C
A04012435    360     72.0     C
A00654400    381     76.2     C
A00577109    419     83.8     B
A00580920    218     43.6     F
A04028610    310     62.0     D
A04063494    460     92.0     A
```

Additional Requirements:

- Your program **must compile** and run, otherwise you will receive a score of 0.
- Your program must pass **Test Case 0** (below) or you will receive a score of 50 or less with no credit for the other grading categories (correctness/constraints/style):

  Input: A00123456 72 72 72 72 0 50 10 1 1 10
  Expected output: See below. File I/O is **not** required to pass Test Case 0 (but IS required for a complete solution to the assignment!)

```
StudentID  Points  Percent  Grade
--------------------------------
A00123456    360     72.0     C
```

- Use an array to store the exam scores and another one for the assignment scores!

- Do not store data for more than one student at a time in your program. (Do NOT use an array to store the data for **all** the students at once).

- The program must be modular, with **at least three functions** in addition to main. Each function should perform a single, well-defined task. Do not write trivial functions such as a function to output a single value.

- Do not use any features of C++ that we have not yet covered in class (use features from Chapters 1-7, 11 only). Do not use classes!

Style:

See the Style Guidelines document on the course website. Especially pay attention to the **comments** required for the top of the file and for functions. The grader will deduct points if your program violates the style guidelines.

Logistics:

Name your file **assign1_xxxxx.cpp** where xxxxx is your TX State NetID (your txstate.edu email id). The file name should look something like this: assign1_js236.cpp

There are two steps to the turn-in process:

1. Submit an electronic copy using the Assignments tool on the TRACS website for this class.

2. Submit a printout of the source file at the beginning of class, the day the assignment is due. Please print your name on the front page, staple if there is more than one page.

See the assignment turn-in policy on the course website (cs.txstate.edu/~js236/cs2308) for more details.