

Programming Assignment #1

Grade Book

CS 2308.251 and 252, Spring 2015

Instructor: Jill Seaman

Due: before class **Friday, 2/6/2015** (upload electronic copy by 9:30am)

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
7 lab scores
7 quiz scores
1 final exam score

example:

A00123456 92 95 97 88 83 84 70 80 90 92 95 97 88 83 84

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 **final average** (computed according to the given weights below) formatted to 1 decimal place, and the **letter grade** (computed according to the grading scale below) **for each student**.

The grading scale:

Score	Grade
89.5 - 100	A
79.5 - 89.49	B
69.5 - 79.49	C
59.5 - 69.49	D
0 - 59.49	F

The weights for each category:

Labs*:	25%
Quizzes*:	25%
Final Exam:	50%
*7 lab scores, 7 quiz scores: the lowest lab score is dropped. the lowest quiz score is dropped.	

Here is an example of how the output should appear.

```
StudentID  FinalAvg  Grade
```

```
-----
```

```
A00529154    85.6    B
A00656624    80.3    B
A02507691    79.8    B
A00612352    78.6    C
A04012435    73.5    C
A00654400    75.3    C
A00577109    69.2    D
A00580920    83.6    B
A04028610    90.0    A
A04063494    90.0    A
```

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 (but **no more than 15% of the points will be deducted for violations to the style guidelines**).

Additional Requirements:

- This program should be developed using a Linux or Unix environment. Start early!
- Your program **must compile** and run, otherwise you will receive a score of 0.
- Use an array to store the lab scores and another one for the quiz 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 only). Do not use structs or classes!

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.

If you are unable to turn a printout in during class, you have until 5pm on the day the assignment is due to turn it in to the computer science department office (Comal 211). They will stamp it and put it in my mailbox. DO NOT slide it under my office door! It must have a timestamp if you submit it outside of class.