

Programming Assignment #7

Keep a Grade Book

CS 1428.253, Spring 2019

Instructor: Jill Seaman

Due: before class **Tuesday, 4/30/2019** (upload electronic copy by 9:00am)

Write a program that manages a grade book for a history course. The program should use a structure that stores the following data:

Student Last Name
Final Average
Letter Grade

The program should create an array of five structures (the roster). The elements should be initialized with the following data:

Student Name	Final Average	Letter Grade
Cooper	0.0	'U'
Fields	0.0	'U'
Garcia	0.0	'U'
Newton	0.0	'U'
Ortega	0.0	'U'

Each time the program runs, it should prompt the user to enter the midterm and final exam score for each student (the student's name should be given as a prompt). After the scores are entered for each student, it should calculate the final average and the letter grade and store them in the array. The midterm counts for 40% of the final average and the final exam counts for 60%. A final average of 89.5 or above is an A, 79.5 and above but less than 89.5 is a B, and below 79.5 is a C. The program should display the name, final average, and letter grade of each student. It should then list the class average, and then the name and final average of the student who scored the highest final average in the class.

Sample output is on the class website.

Additional Requirements:

- Use named constants for the number of students.
- The roster array must be processed using loops.
- Top Down Design: your program must perform these four tasks (write a separate function for each one):
 1. Calculate the final average and letter grade of all the students, based on data input from the user.
 2. Show the student data in a table. Only one digit after the decimal in the final average
 3. Calculate the class average
 4. Determine the student with the highest average
- Since there is only one array in the program you do not need to pass the size as a parameter to the functions (but you can if you want to).

Style:

See the Style Guidelines document on the course website. Especially pay attention to the comments required for functions. **Include the full comments required for your functions this time!** The grader will deduct points if your program violates the style guidelines. Make your program output look nice!

Logistics:

Name your file **assign7_XXXXX.cpp** where XXXXX is your TX State NetID (your txstate.edu email id). The file name should look something like this: assign7_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 (tracs.txstate.edu). Submit the .cpp file, (NOT a .cbp file!).
2. Submit a **printout** of the .cpp file at the beginning of class on the day the assignment is due. Please print your name on the front page.

See the assignment turn-in policy on the course website (cs.txstate.edu/~js236/cs1428) for details, including deadlines, penalties, and where to submit printouts after class.