### Simple Grading Program

Write a program that grades three exams for a class. Your program will read in lines of data that contain the following items:

1. Student ID (an integer)
2. Student name, with the first and last name separated by an underscore, as John\_Cole. Store this as a *string*, not as an array of char.
3. Grades on three tests. These are integers.

As you read each student ID, dynamically allocate a structure that contains student information. Put the pointer that the *new* operator returns into an array of fixed size. You can assume that there are no more than 100 students in any class, but the actual number of “Student” structs you allocate is dependent upon how many you read. So your first step is to read all of the data. While you are doing this, compute the average of the three grades and also store it in the structure. You do not need to check whether there are more than 100 students. Assume that the teacher was careful and there is no bad data in the file.

Having read the file and created the array, display a menu containing three options:

1. Print the list of information in order by name
2. Print the list of information in order of average grade, highest grade first.
3. Look up a student by student ID. Request a student ID and search the list. If the student is found, show all of the information for that one student. If not, show a message.

Printing should show the student ID, the name, the three test scores, and the average. Display of the average should show it to two decimal places, right-justified in a field seven characters wide. Display the name and scores in columns of appropriate width so that everything lines up neatly.

After doing any of the three menu items, return to the menu. Stop when the user enters 0 as his/her choice. Invalid menu choices should show a message, then return to the menu.

Use bubble sort for the sorting. Write this as a single function that can sort by either name or average. You can have a parameter to the function indicating which field is to be used for sorting.

The data file provided is called Asg4Grades.txt. When you open this, just use the file name, no drive letter or subdirectory.

Note that since there are no spaces within the data, you can use stream extraction to read the individual elements.

**To hand in:** Submit only your .CPP file through eLearning. Use the name Asg<NetID.cpp>. If eLearning has a problem with this, put your .CPP file into a Zip file with the same name and submit that. No need to submit the data file.

### Grading

|  |  |
| --- | --- |
| Program meets specifications and works correctly | 50% |
| Program is structured well and uses structs and dynamic allocation | 40% |
| Program comments and variable names | 10% |
|  |  |

Further grading rubric:

Using two separate sorting functions: -20

Using an array of structs rather than an array of pointers to structs: -20

The file name is hard-coded to a specific directory and drive: -25

Sorting doesn’t work: -15

Averages computed incorrectly: -10