

CSCE 1040 J qo gy qtm3 Assignment

You are to write a program that processes grade data using structures. Use `scanf` to input data and `printf` to output the results. You may also use C++ style I/O if you prefer.

You are to work individually.

- Include your name in the code comments. Failure to include your name may lead to a grade of zero.
- Name your file `J y n3.cr r`. Failure to name the file correctly may lead to a grade of zero.
- Submit your program to the Ecpxcu page for the J qo gy qtm3.
- The input data will be a data file that you redirect into your program. The data file can be found at `~dmk0080/public/1040/j y n3qpg/versionA/grades` on the CSE server.
- The assignment must be submitted by the f vg'f cvg'kugf 'lp'Ecpxcu.

There are files to get you started on this assignment located in `~dmk0080/public/1040/j y n3qpg/versionA/`

grades - the data file

student.h - structure to hold the student information

bubble.h - bubble sort function prototype

bubble.cr r - bubble sort that sorts an array of student pointers

******j y n3qer r* - a main program to get you started

Your program is to use two data structures to read in student grade data, perform some calculations, sort the students in ascending order by average, determine some class statistics, and output the results. The first data structure is `classStats` and should have variables `mean` (float), `min` (float), `max` (float), `median` (float), and `name` (character pointer). You will need to create this structure yourself, placing it above `main()` in *lab8assignmentA.c* and create one variable of type `classStats` (not a pointer) in your main program.

The second data structure is called `student` and will have variables `first` (character pointer), `last` (character pointer), `exam1` (integer), `exam2` (integer), `exam3` (integer), and `mean` (float). This structure is in the file *student.h*. Take a moment to study this structure to understand it before using it. You will need to create an array of 19 `student` pointers and will need to allocate space for each in your main program using `malloc()`.

The data file contains the name of the course followed by 19 students, each student having three exam grades. Use the array of `student` pointers to store the information as read in using `scanf`. An example data file is below:

```
CSCE1040
Erica Sanders 75 89 67
Kelley Cummings 74 70 79
Jamie Reynolds 64 52 66
Shawna Huff 80 88 61
Muriel Holmes 81 74 79
Marion Harmon 77 64 69
Catherine Moss 51 80 73
Kristin Fernandez 86 69 81
Elsa Alvarado 63 77 67
Cora Spencer 76 79 71
Valerie Olson 85 78 79
Anne Singleton 85 87 65
Rene Boone 85 85 77
James Morgan 69 86 51
Cedric Haynes 72 73 88
Elijah Snyder 65 92 91
Roger Howard 79 95 71
Archie Black 70 81 63
Melvin Watkins 66 67 72
```

Use the three grades to determine the student's mean and store it with the name and exam grades in `student` structure. Assumed each exam is weighted the same. Once all the students are read in and the averages determined, sort

the students using the bubble sort code in *bubble.c*, which takes an array of `student` pointers and the array size as arguments. Take a moment to study *bubble.c* to understand how it works before using it. After sorting the students based on mean, find the mean, minimum, maximum, and median of the grades and store them in the `classStats` structure.

Use `printf` to output the data. Your output should appear as below (include the line of digits), which displays the class statistics and the student averages. Do not worry about rounding. If some students are out of order because they have the same mean, do not worry about it as long as you used the bubble sort.

```
123456789012345678901234567890123456789012345678901234567890
CSCE1040 MEAN:  74.91 MIN:  60.66 MAX:  82.66 MEDIAN:  76.33
    Jamie      Reynolds  60.66
Catherine      Moss    68.00
    Melvin     Watkins  68.33
    James      Morgan  68.66
    Elsa       Alvarado 69.00
    Marion     Harmon  70.00
    Archie     Black   71.33
    Kelley     Cummings 74.33
    Cora       Spencer 75.33
    Shawna     Huff    76.33
    Erica      Sanders 77.00
    Cedric     Haynes  77.66
    Muriel     Holmes  78.00
    Kristin    Fernandez 78.66
    Anne       Singleton 79.00
    Valerie    Olson   80.66
    Roger      Howard  81.66
    Rene       Boone   82.33
    Elijah     Snyder  82.66
```

Hint 1: Break the programming into smaller phases. Read in the data and output it to make sure it works. Then add code to calculate the student averages. Next, add code to sort the students using the bubble sort. After that, add code to determine the class statistics. Finally, output the results.

Hint 2: To compile your code, you can use `g++ *.crr` to compile all `.c` files in your directory at one time.