CS 135 - Computer Science I
Fall 2021
Final (`final.cpp`)

Code Limitations

Please refrain from using advanced C++ code techniques. Limit commands to those discussed in class and covered in chapters 1-9. You may use the following resources:

- your text
- you online syllabus and links therein
- past program submissions
- Bellagio
- code that you have authored for this course

!! **If you are caught cheating you will receive a zero (0) grade for the entire final exam.** !!

Due Date

You must complete this assignment before the end of the day on the date shown below. No late submissions will be accepted.

Section	Due
All	Dec 17 ~5:59 pm

Assignment

This assignment assumes that you have understood and completed all the other assignments in the course. You will be drawing on your previous experience in understanding directions and writing code to complete the final exam. This exam tests students on their practical skill in reading instructions, modifying existing algorithms, and crafting a C++ program that is properly formatted and implements proper commenting rules and syntax.

Create a program that uses a single array of structures. You will be using the data file: `Using the data from /home/shared/cs135/dbrodersen/final/data.txt`. your program shall output details about persons with the highest average and the lowest average in the class as exemplified in the sample output.

If you have read your textbook, *Figure 1* will be meaningful and very useful as you code your program.

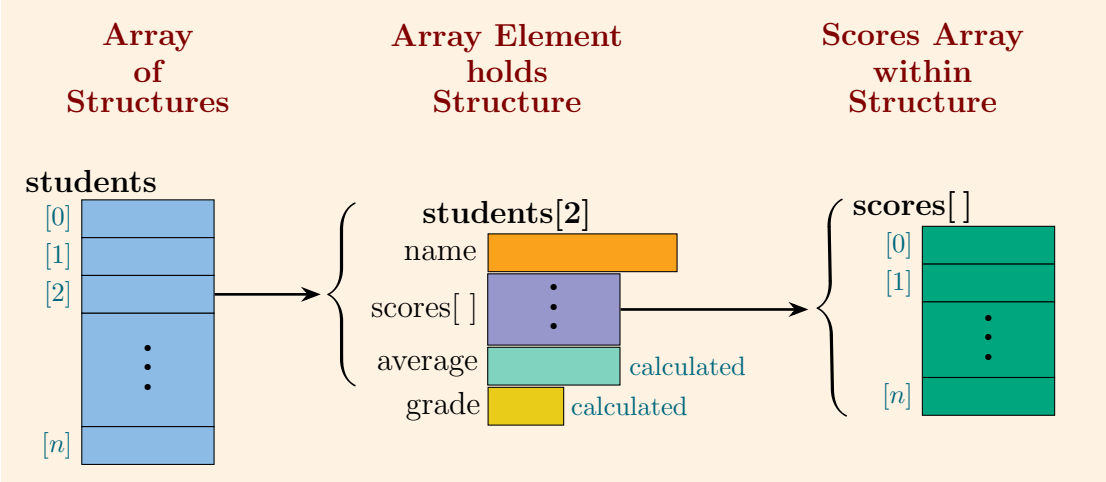

Figure 1: Data Organization

Minimum Specifications

!! Use a modular design for this program. You must use functions !!

- Your program must compile and run on Bellagio with a minimum number of warnings
- You will receive no credit for the program if it will not compile
- You will receive no credit for the program if it will not compile or run
  - ☐ You must have one array to hold structures
  - ☐ You must define at least one structure with the name `student_t`
  - ☐ Your program must have at least the following Functions:
    - ▷ *getData* to get the data into your variables
    - ▷ *calculateAverage* to calculate the average for each student
    - ▷ *calculateGrade* to calculate the grade for each student
    - ▷ *findMaxAve* to find the maximum average grade
    - ▷ *findMinAve* to find the minimum average grade
    - ▷ *printReport* to print student a report as exemplified below
- Your program must follow the coding standards for this course. You cannot earn anything above an 80% if you fail to **format** your code correctly.

- Use proper code commenting. You cannot earn anything above an 80% if you fail to **comment** your code sufficiently or correctly.
- Using the data from **/home/shared/cs135/dbrodersen/final/data.txt**, your output should mirror the expected output provided in *Figure 2*.

Expected Output

```
*** line space here***
Student Grade Report
-------------------------------------------------------------------
Name        Test 1  Test 2  Test 3  Test 4  Test 5   Average   Grade
-------     ------  ------  ------  ------  ------   -------   -----
Johnson         85      83      77      91      76     82.40     B
Aniston         80      90      95      93      48     81.20     B
       ...
       ...
Sunny           79      85      28      93      82     73.40     C
Smith           85      72      49      75      63     68.80     D
-------------------------------------------------------------------
Class average:   70.94
-------------------------------------------------------------------

Best Performance:
-------------------------------------------------------------------
Name        Test 1  Test 2  Test 3  Test 4  Test 5   Average   Grade
-------     ------  ------  ------  ------  ------   -------   -----
Bronson         93      94      89      77      97     90.00     A
-------------------------------------------------------------------

Poorest Performance:
-------------------------------------------------------------------
Name        Test 1  Test 2  Test 3  Test 4  Test 5   Average   Grade
-------     ------  ------  ------  ------  ------   -------   -----
Blair           23      45      96      38      59     52.20     F
-------------------------------------------------------------------
*** line space here***
```

Figure 2: Expected Output

Turn in final

This assignment will NOT be turned in using Canvas. Use the command below while logged into *Bellagio* to turn in your assignment. Note: you must be in the directory where final exists for this command to work.

Bellagio Linux Console (Terminal)

```
$ turnin -c cs135-dbrodersen -p final -v final.cpp

— results of a successful submission —
Submitted files:
final-username
final-username/final.cpp
 $
```

The `turnin` command submits assignments to be graded. The drop box for the assignment closes promptly at the due date/time. If the project is enabled and you want to resubmit your assignment, you may do so by re-running the `turnin` command. **New submissions overwrite any previous submissions.**

Syntax: `turnin -c` *course-name* `-p` *project-id* `-v` *file name(s)* separated by spaces.

Option	Purpose
`-c`	Course. Sets the course to which we'll submit our assignments. Note that the course includes your registered section number. *Required.*
`-p`	Project. Sets the project to which we'll submit our assignments. The project idea is always shown at the top of each assignment, e.g. `pa02`. *Required.*
`-v`	Verbose. Prints a list of submitted files once they have been submitted.
`-h`	Help. Print a help message. You can also type `man turnin`.
`-l`	List. Prints a list of projects, along with whether or not they are enabled and shows which project is the default project.