**CS 102 Fall 2021 Lab 4: Team Project**

Reminder: late submissions will not be accepted

Each person needs to submit. Hopefully your submission will match your teammates' submissions. Provide the following:

* List all team members, including yourself.
* Team name
* Describe expectations of each team member (get an A, pass the class, have fun, etc.)
* Describe how the team will communicate with each other
* Describe the days/times during which you will work together
* Describe each person's specific strengths and expected contributions

**3 Assignments**

Team membership and all 6 bullet points above: November 20, 11:59pm (10 points)

Live demo/presentation in class December 7, 6:30 pm (20 points)

CANVAS Report Part 2 and source file submissions Dec 9, 11:59pm (70 points)

**Purpose**

This lab assignment is to provide you with an opportunity to:

* demonstrate your ability to write a C++ program utilizing basic C++ concepts (specific concepts should be listed as part of your report)
* demonstrate understanding of concepts covered in the course
* work with 2-D arrays of struct’s.

**Lab Requirements**

* This lab assignment requires submission of source file(s) and a lab report. For full credit for this lab, these following files must be submitted via Canvas by the assigned due date:
* a lab report, which meets the requirements as described below (pdf).
* your source file(s) (cpp)

**Description**

Write a program that will read the data from the supplied input file SuperBowl.txt, this is in Canvas under “Files”. There will be two main data structures for you to store the data in. The first is a struct with just two members: the team name, and the score.

The second main data structure you will use is a two-dimensional array of these struct. If you view this as a table there will be two columns. The first column is a struct for the Super Bowl winner’s team name and the winner’s score. The second column is a struct for the Super Bowl loser’s team name, and score. Treat the number of rows as unknown, as it will increase every year when there is a new Super Bowl, but it is greater than 1 and less than 100.

1. You may assume that every row of input contains four values:

#1 is the name of the winning team name (note that this might or might not contains spaces, that is it might be one or more words).

#2 is the score of the winning team.

#3 is the name of the losing team name (note that this might or might not contains spaces, that is it might be one or more words).

#4 is the score of the losing team.

1. The number of rows is unknown, as the number of Super Bowls will increase over time.
2. Below is some example input, with the real team names and scores for the first 4 Super Bowls. Note that a real SuperBowl.txt file is included as input. It is in Canvas under “Files”.

Green Bay 35 Kansas City 10

Green Bay 33 Oakland Raiders 14

New York Jets 16 Baltimore Colts 7

Kansas City 23 Minnesota 7

1. Use separate functions for each of the following:
2. Compute the average winning team’s score
3. Compute the average losing team’s score
4. Compute the maximum score (highest score from any team in any Super Bowl)
5. Compute the minimum score (lowest score from any team in any Super Bowl)
6. Print out 9 lines of output, that closely resemble below, with **<BLAH>** replaced by your computed values:
   * + First Super Bowl winner: **<BLAH>**, score: **<BLAH>**
     + Most Recent Super Bowl winner: **<BLAH>**, score: **<BLAH>**
     + Maximum score from any Super Bowl team: **<BLAH>**
     + Minimum score from any Super Bowl team: **<BLAH>**
     + Average winning score from Super Bowl teams: **<BLAH>** *For average make it a double with 2 digits after decimal.*
     + Average losing score from Super Bowl teams: **<BLAH>** *For average make it a double with 2 digits after decimal.*
     + The team(s) that have won the most times in a row are: **<BLAH> <BLAH> <BLAH>** they won **<BLAH>** times in a row. Replace **<BLAH> <BLAH> <BLAH>** *with team name(s) – example above if it is a three-way tie.*
     + The team(s) that have lost the most times in a row are: **<BLAH> <BLAH> <BLAH>** they lost **<BLAH>** times in a row. *Replace* ***<BLAH> <BLAH> <BLAH>*** *with team name(s) – example above if it is a three-way tie.*
     + The team(s) that have never won but lost the most super bowls are: **<BLAH> <BLAH> <BLAH>** they lost **<BLAH>** times. *Replace* ***<BLAH> <BLAH> <BLAH>*** *with team name(s) – example above if it is a three-way tie.*

Take screenshots of your program during development and testing. These may be handy for your report.

**Lab Report requirements**

The lab report should have these required sections:

* Purpose
* Planning and organization
* Development process
* Product
* Pitfalls
* Possible improvements

The lab report should be submitted as a .pdf in Canvas. You may create the report in a document creation software of your choice. Please ensure that your lab report is cleanly formatted and free of distracting grammatical or other errors. These errors will cost you points on clarity. For more specifics, check Canvas for the grading rubric.

**Purpose**

Provide a few sentences describing the purpose of this assignment, not just the purpose of the program. ( You may include a description of what your program does. But this is different than why we are doing this assignment.) Please use your own words. (I don’t need to read my own words again.) This should be in a short paragraph form. (A paragraph includes complete sentences, not a collection of phrases like a series of text messages.)

**Planning and organization**

Describe the process you followed and/or will follow in order to plan out and structure this project. The steps here should be specific enough for others to replicate your process. Provide screenshots, other visuals, urls, etc. as needed. You do not need to reproduce the assignment directions, but provide other information to allow another to be successful in this assignment. All visuals should be titled and discussed, i.e. not just stuck on the page with no context.

This should be mainly in paragraph form, with lists of steps and visuals as needed. Use your own words. The description does not need to include every detail, but, as mentioned above, should provide enough information so that your planning process can be replicated.

Include a timeline to show your planned activities, what has been accomplished, and the expected timepoints for other activities to lead to the successful completion of your assignment.

At least one flowchart to show program flow is required. The flowchart must have a start and an end. All decision points should be represented as a diamond.

**Development process**

Describe the process you followed in order to develop the code for this project. Which parts did you choose to develop first? Why? How did you use the plans you developed? How did you test your program to make sure it works correctly?

This should be mainly in paragraph form, with lists of steps and visuals as needed. Use your own words. The description does not need to include every detail, but, as mentioned above, should provide enough information so that your planning process can be replicated.

**Product**

Describe your program run(s). Include screenshots for clarity. Discussions should include how the test cases show the capabilities and limitations of your program. See the rubric for more details.

**Pitfalls**

Describe any difficulties and issues you encountered during this assignment. Provide screenshots and other visuals as needed to describe these clearly. Describe how you resolved these issues. If somehow, you didn’t have any difficulties, issues, programming errors, or bugs, include an explanation of why this could be the case. See the rubric for more details.

**Possible improvements**

What could you have done differently to improve your completion of this assignment? Be specific. What could be improved about this assignment overall for you and/or for future students? Be as specific and constructive as possible. See the rubric for more details.

**Program file submission requirements**

Submit the files via Canvas

Programming practices expected for this course:

* good comments to identify the programmer and the project
* good comments for any significant blocks/lines of the program, comments which explain the purpose of the code and not just what the code does
* professional and user-friendly interactions and output