COMP 1130 Principles of Programming II
Project 6 – Falcon Hoops
Maximum Possible Points: 20

Due Date:

- The In-Lab part of the project needs to be demonstrated by the end of the lab period.
- The complete source code and one program execution screenshot needs to be submitted by the beginning of next lab period.

Objectives:

- To gain experience with structures, sorting and searching, and file parsing.

Overview:

For this project, you will have to write a C++ program to compute some statistics about the 2022-23 Flacon Hoops season. You should define a structure type called *Player* and compute some player statistics. Specifically, for each player, it should compute field goal percentage, three pointer percentage, free throw percentage, and average points per game. It should read the data from an input file, perform the calculations, and display the result to the screen. Your program should also allow one to search a particular player and display his numbers. The expected execution trace for the correct program is shown in *Sample Execution*. The data file will contain each player's records. The file will be formatted as follows:

<Name>: *<GamesPlayed> <FieldGoalsMade> <FieldGoalsAttempted> <ThreePointersMade> <ThreePointersAttempted> <FreeThrowsMade> <FreeThrowsAttempted>*

You will need to declare a structure called *Player*. The structure should have several members for each player: name, number of games played, field goals made, field goals attempted, three pointers made, three pointers attempted, free throws made, free throws attempted, field goal percentage, three pointer percentage, free throws percentage, and average points per game. The structure should have a 0-argument constructor that will initialize all the members to 0.

You will need to create an array of *Player* structures in main to store the player data. Your program should have the following functions in addition to main:

- The *readData* function should accept the structure array. It should ask the user for the input file name, open the input file, and read each player's data from this file and store the data in the structure array. This function should also return the number of players. If the file cannot be opened for some reason, the program should display an error message and exit.
- The *calculateStats* function should accept the structure array and the number of players. For each player, tt should calculate the filed goal percentage, three pointer percentage, free throw percentage, and average points per game, and then save them in the appropriate member of the structure. Note that the number of field goals include the number of three pointers as well.

- The *sortPlayers* function should accept the structure array and the number of players, and sort the players in descending order of their average points per game. You may use any of the sorting methods discussed in class.
- The *displayStats* function should accept the structure array and the number of players, and display the players' statistics in the format shown in the Sample Execution.
- The *searchPlayer* function should accept the structure array and the number of players. It will ask the user to enter the name of a player and will display his statistics in the format shown in the Sample Execution.

Instructions:

- This will be an individual programming project.
- Write your code in a file called **prog6.cpp**.
- Use meaningful variable names, helpful comments, and a consistent coding style.
- All program files should have the appropriate comment block at the top:

```
// Name: Your Name
// File Name: prog6.cpp
// Date: Day Month, Year
// Program Description: brief description of the program
```

Deliverables:

You will need to submit the following in Blackboard by the beginning of the next lab period:

- The complete source file and the screenshot of your program execution in Visual Studio or Eclipse.

Grading:

This project is worth 20 points distributed as follows:

In-Lab Demo (10 pts)
Correct declaration of Player structure (2 pts)
Correct implementation of readData (3 pts)
Correct implementation of calculateStats (3 pts)
Correct implementation of displayStats (2 pts)
Complete Program (6 pts)
Correct implementation of sortPlayers (2 pts)
Correct implementation of searchPlayer (2 pts)
Correct implementation of main function (2 pts)
Program Style (4 pts)
Meaningful variable names (1 pt)
Proper indentation (1 pt)
Sufficient comments (2 pts)

YOU WILL LOSE 50% OF THE POINTS YOU RECEIVE IF YOUR PROGRAM DOES NOT COMPILE AND YOU DO NOT SUBMIT THE SCREENSHOT.

Sample Execution:

```
Please enter the input filename: hoops.txt


Name               FG%    TP%    FT%    Average
-----------------------------------------------
Isaiah Sanders     0.47   0.43   0.87     18.6
George Mangas      0.59   0.33   0.73     14.6
Briggs Parris      0.46   0.39   0.94     12.2
Zyon Dobbs         0.45   0.38   0.71     11.8
Fonz Hale          0.39   0.33   0.80     11.7
Tariq Woody        0.63   0.28   0.69      8.4
David Jolinder     0.55   0.52   0.81      8.3
Seth Younkin       0.49   0.00   0.56      3.0
Przemyslaw Golek   0.56   0.50   0.00      2.5
Adam Deininger     0.80   0.78   0.00      2.4
Tommy Williams     0.33   0.29   0.84      2.2
Tyheil Peterson    0.63   0.17   1.00      1.9
Zac Kimball        0.57   0.57   0.00      1.3
Tyler Ganley       0.43   0.39   0.00      1.1
Trey Washenitz     0.33   0.33   0.00      0.8
Colin Comer        0.00   0.00   0.00      0.0

Do you want to search any player? (y/n) y
Enter the player name: Zyon Dobbs

Number of games played: 32
Field goal %: 0.45
Three pointer %: 0.38
Free throw %: 0.71
Average points per game: 11.8

Do you want to search another player? (y/n) y
Enter the player name: Isaiah Dobbs
Player not found or you did not spell the name correctly.

Do you want to search any other team? (y/n) n

Bye!
```

** Anything typed in **blue** indicates a user input.