CS 2 – Homework 2 Possible Points: 110

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**NBA Players’ Stats Version 2**

Using object oriented design write a C++ program that keeps track of professional basketball players’ statistics addressing the following software requirements at a minimum.

1. Program must use a header file to define **Player** **class. (10 points)**
   1. Using private members, Player class should store each player’s stats such as: player’s name, team name, points, rebounds, and assists.
   2. Using public members class should contain prototypes of required constructors and destructs, getter and setter functions and other operational functions as necessary to work with the class data members.
2. Implement class methods/member functions in a separate .cpp source file. **(25 points)**
3. Program should store up to 100 players’ records.
4. Using a function program must read the existing players’ stats from players.txt file – see the sample file provided. Data value is stored in the following order for each player: player’s name, team name, points, rebounds, and assists on each line. **(10 points)**
5. Program must be menu driven, giving user various choices such as: add a new player, display all players’ data, display individual profile data and edit specific player’s data (team, points, rebounds, assists). **(20 points)**
6. Program must use a function to display all players’ stats in a clean and nicely formatted tabular form. **(10 points)**
7. Program must use functions to search array to find the index of a specific player which can be then used to display, edit and update the data of the existing player. **(10 points)**
8. Program must use a function to add a new player’s information. **(10 points)**
9. Program must give user option to save data into the same players.txt file. **(5 points)**
10. **Bonus 10 points.** Give user option to display the players’ records in descending order based on player’s average points.

**Other requirements:**

1. Write date, programmer and brief program information as comments at the top of all the source files.
2. Follow best programming practices:
   1. Write adequate comments. Comments can help you plan for the solution and think through the problem.
   2. Do NOT use global variables; constants are okay.
   3. Use indentation and whitespaces to cleanly format your code.

**Deliverables:**

1. **Upload all the source file(s) \*.cpp, \*.h, etc., into D2L Homework 2 dropbox before the due date.**

Note: You may submit partial codes for, well, partial credit.