CS221: Data Structures

Programming Assignment #2

Due – See canvas for due Dates/Time (20 points)

PROGRAMS ARE INDIVIDUAL WORK ONLY. Do not collaborate on the solutions.

Problem Statement

You are to develop a program to read Hockey Player objects from an **input file**. You may reuse your player object from Program 1. See that assignment for the requirements for a single player. We are adding additional computations to this program.

Each line of the data file will contain one player's name and stats. The format for a line of player data is the same as before:

firstname lastname age shotsTargetFor shotsTargetAgainst missedShotsFor missedShotsAgainst met bodyWeight

The statistics are integers and must be stored in a single array within a Player object.

You may assume there are no errors in the data. I will not leave data out or put non-numeric data in numeric fields. You must read until the **end of the file** is found.

New Requirements

You are to implement an ordered **PlayerList** data structure/data type that stores players using a dynamically allocated ***array*** to store the data within the class. Add players to the List based on their lastname, firstname (use the firstname to break ties in sort order).

You must implement the following operations on your List along with any other utility functions you might need. You may also need to add operations to your Hockey Player class, as well.

Operations for PlayerList

1. **Default Constructor –** to create the list with a capacity (max size) passed in by the user
2. **Add –** Add a player to the List; players should be added so that they are maintained in order.
3. Iterate through the List so that you can get each item out for printing
    a. You must implement **hasNext**() and **getNext**() on your list for this to work.
4. **Clear** out the list to make it empty
5. Test if a list **isEmpty**()
6. Get the
    a. **size of the list**,
    b. **total calories burned by the players** in the list

SUMMARY OF OPERATION:

1. Prompt the user for the input and output file names. DO NOT hardcode file names into your program.
2. Open input file
3. Read each player and add them to your List.
4. Keep track of the number of items in the list
5. Open an output file
6. Write a report summary line to the output file, then
7. Write each item from the list into the output file, along with any other output required by the assignment (see sample report at end of document).

OTHER REQUIREMENTS:

1. Modify the method that reads a player data from the console (cin) to use an istream as a parameter for reading from any input stream such as a data file.
2. Split your program code files into the appropriate .cpp and .h files for each of your classes.
3. Name your project's main file **Program2.cpp**

TURN IN:

1. Submit a single file with your .cpp and .h zipped up into it.
2. Please also include the examples of your input test files

GRADING REQUIREMENTS:

1. Your program **must be well-commented**. Comment all variables, functions and remember to have a section of comments at the top of your program that includes your name, date, course section and a description of what your program does. *(Internal documentation on programs in my course counts for up to 20% of credit).*
2. Use good variable names
3. Use good and consistent naming conventions for class members.
4. Use proper code indentation to make sure your program is easy to read and understand.
5. You will receive **no more** than 50% credit if your program does not compile.
6. If your program compiles but does not execute correctly, you will receive no more than 70% credit.

SAMPLE EXECUTION:

```
Welcome to the hockey player statistics calculator test program.
I am going to read players from an input data file. You will tell me the names of your input and output files.
I will store all of the players in a list, compute each player's stats and then write the resulting team report to your output file.

Enter the name of your input file: playerinput.txt
Enter the name of your output file: report.txt

Reading Players from: playerinput.txt
The data has been written to you output file: report.txt

End of Program 2
```

SAMPLE INPUT FILE AND CORRESPONDING OUTPUT FILE:

1. Sample input text file -

firstname lastname age shotsTargetFor shotsTargetAgainst missedShotsFor missedShotsAgainst met bodyWeight

```
Axel Toupane              24  3651 1390 1055 1290 2032 144
Bobby Portis              30  1510 250  1764 720  365  198
Brook Lopez               33  2174 1121 2987 1931 453  210
Bryn Forbes               30  1127 138  1318 484  286  204
Donte DiVincenzo          23  1068 1281 1806 980  371  202
Elijah Bryant             21  2005 235  2140 1253 220  161
Giannis Antetokounmpo     26  4391 303  6342 1080 202  265
Jeff Teague               25  2313 1019 1323 1191 169  216
Jrue Holiday              35  3899 231  3994 926  590  170
Khris Middleton           33  4524 750  5421 1700 269  219
Mamadi Diakite            27  1106 236  1709 787  491  230
P.J. Tucker               33  2982 1000 2983 473  248  214
Pat Connaughton           28  1195 545  4581 1019 154  232
Sam Merrill               23  1771 512  1086 1510 272  181
Thanasis Antetokounmpo    34  1074 100  2055 670  187  170
```

Note: There are no blank lines at the end of the input data files

2. Sample output file -

```
BASKETBALL TEAM REPORT --- 15 PLAYERS FOUND IN FILE
TOTAL CALORIES BURNED: 15702.6

    PLAYER NAME          :      FF%        Calories burned
-------------------------------------------------------------
 Antetokounmpo, Giannis :      114.7      936.7
Antetokounmpo, Thanasis :      132.6      556.3
         Bryant, Elijah :      156.0      619.8
       Connaughton, Pat :      137.5      625.2
        Diakite, Mamadi :      157.0      1976.2
      DiVincenzo, Donte :      468.8      1311.4
           Forbes, Bryn :      134.1      1021.0
          Holiday, Jrue :      117.1      1755.2
           Lopez, Brook :      244.7      1664.7
           Merrill, Sam :      342.1      861.5
       Middleton, Khris :      132.6      1030.9
          Portis, Bobby :      142.1      1264.7
           Teague, Jeff :      254.9.1    638.8
          Toupane, Axel :      232.2      511.5
           Tucker, P.J. :      132.7      928.7
```