PROGRAM 09-1 LINKED LISTS

This is a significant programming assignment to test your ability to develop and use doubly-linked lists of non-trivial objects.

PROBLEM STATEMENT

You are to develop a program to create a sorted *doubly-linked list* class that holds Baseball Players. You will read the players from the file, one at a time, and add each one to your List. See Program-06-1 for the requirements for a single player.

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 Baseball Player class, as well.

Operations for PlayerList

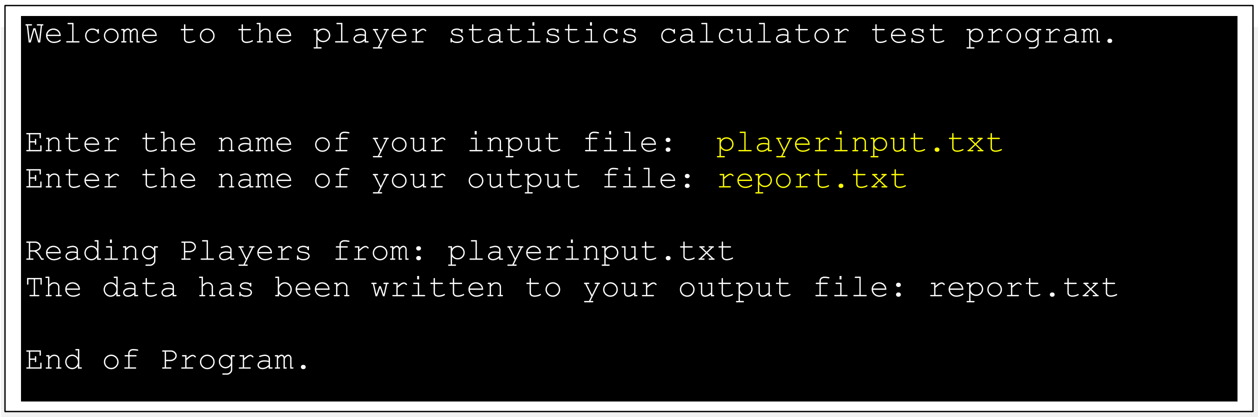
- Default Constructor
- Destructor that properly frees the memory allocated to the linked structure
- Add a player to the List – maintain the players in the list by adding them in the correct location by using their name for sorting. (Sort in ascending alphabetical order by last name, use first names if needed to break ties.) You may ignore case issues for this program.
- Iterate through the List so that you can get each item out for printing
    - You must implement some form of iterator using hasNext() and getNext() on your list for this to work.
- Clear out the list to make it empty
- Test if a list isEmpty()
- Get the size of the list
- Compute the team's **overall batting average** as an average of the player's individual averages.

SUMMARY OF OPERATIONS

- Prompt the user for the input and output file names. DO NOT hardcode file names into your program.
- Open input file
- Read each player and add them to your List.
- The List itself should keep track of the number of items stored in it
- Open an output file
- Write a report summary line to the output file, then
- Write each item from the list into the output file, along with any other output required by the assignment
- remove is not required for this program, but it is a good exercise to see if you can implement It properly.

Make sure you test your list to ensure the forward AND backward links are set properly. One way to do this is to iterate through the list in reverse. If your data is visited/printed correctly, it is an indication your back links are properly implemented.

SAMPLE EXECUTION



```
Welcome to the player statistics calculator test program.


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 your output file: report.txt

End of Program.
```

SAMPLE INPUT FILE

```
Hank Aaron     13941 12364 2294 624  98 755 1402 32
Chipper Jones  10614  8984 1671 549  38 468 1512 18
Ty Cobb        13099 11434 3053 724 295 117 1249 94
Jonny Bench     8674  7658 1254 381  24 389  891 19
Tony Gwynn     10232  9288 2378 543  85 135  434 24
John Smoltz     1167   948  118  26   2   5   79  3
```

Note – there should be NO blank lines after the last line of data in an input file, or extra blanks at the end of lines.

SAMPLE OUTPUT TEXT FILE (NOTE SUMMARY AT TOP!)

```
BASEBALL TEAM REPORT --- 6 PLAYERS FOUND IN FILE
OVERALL BATTING AVERAGE is 0.290

    PLAYER NAME      :    AVERAGE    OPS
---------------------------------------------
         Aaron, Hank :     0.305    0.928
        Bench, Jonny :     0.267    0.817
            Cobb, Ty :     0.366    0.934
         Gwynn, Tony :     0.338    0.810
      Jones, Chipper :     0.303    0.930
        Smoltz, John :     0.159    0.406

For testing, list in reverse order is:
    PLAYER NAME      :    AVERAGE    OPS
---------------------------------------------
        Smoltz, John :     0.159    0.406
      Jones, Chipper :     0.303    0.930
         Gwynn, Tony :     0.338    0.810
            Cobb, Ty :     0.366    0.934
        Bench, Jonny :     0.267    0.817
         Aaron, Hank :     0.305    0.928
```

TIP

Work the problem in stages. First work the list as a singly linked list, then when that is operating correctly, go back and add in the logic necessary to handle the backward links.

REFERENCES

I am using the online baseball reference (www.baseball-reference.com/players) to look up batting statistics. Please note that the on-base percentage I get is slightly off of the actual percentages due to not using quite ALL of the plate appearance data in my calculations.