CS221: Data Structures — Programming Assignment #1

Due: See Canvas for Due Date/Time — (10 points)

Problem Statement

You are to develop a program to define and test an object class whose instances will be used to store some batting data for a baseball player. We use this information to compute some statistics about our baseball players. A baseball player will have the following data:

firstname lastname plateappearances atbats singles doubles triples homeruns walks hitbypitch

The 8 statistics are integers and must be stored in a single array within a Player object.

Your class needs to provide all the methods needed to:

- provide a default constructor for your Player class
  - default first and last name are "unknown"
  - default integer values are 0
- read a player's data from a single line of input
  - These will be used later when we start reading records of data from an input file. The next iteration of this program will read a player from a single line of input file, so we want to set this up to make that easier to reuse.
  - You will pass a parameter that is a referenced to an istream. This function will assume the next line of input on that stream is valid. We will use cin as the stream to test it with. I will discuss this during the C++ review in class.
  - The reader method MUST NOT print messages to or prompt the user for data. It is a utility only.
- Get methods for any data the test driver program might need to display on the screen later (at a minimum gets to retrieve each name and the values needed for this program's output)
- compute the Player's Batting Average and OBP values*
- Any other methods you need to solve the problem.

TEST DRIVER

Once you have developed a good class definition, you need to write a "test" main program to test it out. Your test program should declare a variable of type Player and then prompt the user for the data to be set into the object. Use the player class reader method to retrieve the values from the input stream. Then, display the characteristics of the Player for the user to the screen.

Note that all interactive prompts are coming from the main program. The Player object itself is passive - it is not doing the prompting for values. All values are read into the object by using its methods.

The pages at the end of this document has a sample test that I ran with my program. Yours should behave and look the same way. (My sample input is highlighted in yellow).

Note that the value of this program is 10 points (*out of a projected total of 100 programming points*). That is because we are essentially defining the beginnings of a class that we will enhance over several programming assignments. In other words, follow on programs will build on this one. As they become more difficult/complex/time-consuming, they will be worth more points.

*Computing Averages:

Batting Average:

The batting average is the sum of the hits (singles, doubles, triples, home runs) divided by the number of at-bats.

On Base Percentage:

We are also going to compute on base percentage. OBP is the sum of all hits, walks and hit by pitch divided by the number of plate appearances. Please note, if you are using the official baseball references, this computation is slightly off because there are a few mores stats actually used in the official statistic.

We will add other computations in the next version of the program.

Other Requirements:

- Please make sure that you create **Console Application** in Visual Studio 2019 or comparable compiler. If you are using another system IDE, I will need you to use ANSI C++11 version of the compiler. Also, watch for instructions on how to submit these programs to canvas.
- ***Include in the comments at the top of the main program file the version of the compiler you are using. Don't forget your name/date/etc.***
- Make sure when you begin, you create an **empty** project, with no pre-compiled header.
- Name your project **Program1**.
- You may never use built-in C++ template libraries or smart pointer data types in this course unless instructed to

TURN IN:

- Submit a single zipped file of your project source code to canvas for this assignment. Include only the source code files in the zip (.cpp, .h) and ensure that no hidden directories or other support files are included in the zip.
- If your program is in a single code file only, you may just upload that file.
- You may include the Visual Studio project files or linux makefiles but do not include any of the subdirectories such as Debug or .vs in the upload.
- *If you used special settings or compile switches, please include that information in the comments at the top of your main program file.*

Grading Requirements

- 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 courses counts for up to 20% of credit.)*
- Use good variable names.
- Use good and consistent naming conventions for class members.
- Use proper code indentation to make sure your program is easy to read and understand.
- You will receive no more than 50% credit if your program does not compile.
- If your program compiles but does not execute correctly, you will receive no more than 70% credit.

Sample Execution:

```
Welcome to the baseball player statistics test driver program.
When prompted, please enter the player's data in the form of
firstname lastname plateapps atbats singles doubles triples homeruns bbs hbp.
For example:

Chipper Jones 10614 8984 1671 549 38 468 1512 18


Enter Player Data: john smoltz 1167 948 118 26 2 5 79 3
    john smoltz's Batting Average =  0.159 and OBP =  0.200

Do you wish to test another [y/n]? y

Enter Player Data: chipper jones 10614 8984 1671 549 38 468 1512 18
    chipper jone's Batting Average =  0.303 and OBP =  0.401

Do you wish to test another [y/n]? n

Program 1 Testing Complete
```