CSC 2220 – Programming in Java
Fall 2020 Semester, Block A
Programming Assignment 2
Due Date: Wednesday, September 9, 2020, by 11:55pm, to Moodle

General Rules of Engagement

Unless otherwise stated, you are not to collaborate on this, or any, assignment. Refer to the Cheating Policy in the syllabus for more information.

When you turn in on Moodle, you'll probably have several Java classes to submit. Submit everything via .zip file. All of the .java files that are necessary.

Although you won't necessarily see a deduction, do take care to make sure spelling and spacing have been addressed and are correct.

Make sure to comment your code. You should have a header for each of your .java files, which includes name and description of what the class does. You should also have comments throughout your code. Failure to do so will result in a loss of points.

No late work will be accepted. If you are up against a deadline and Moodle isn't responsive for some reason, remember you can always email your zip file to me, at rjh7g@mcs.uvawise.edu for no penalty. Just remember to use common sense.

Make sure that you name things meaningfully. This is just good programming practice and a way to self-document your code beyond writing comments.

Programming Assignment Description

In lectures, we've been working on digitizing the wooden baseball game I showed you at the beginning of the semester. Now, between the first assignment, and what you have done in class, we should be able to start working with line-ups for home and away and track stats.

As opposed to the first assignment, let's not keep score yet. Let's practice recording the results of plays as appropriate stats for a batter in the line-up.

Assumption: Let's be nice and implement just like the rules this year – a universal DH for each side. If you're unfamiliar, typically there are nine batters (and positions) in the game. If we were playing in a typical American League year, the eight positions are the other positions in the field besides the pitcher; the additional position is known as the designated hitter position (DH). In the National League, prior to this year they did not have a DH; the pitcher was used instead.

You'll need to come up with modifications to the Batter class we began working on during lecture on Wednesday. You will find the attributes and methods necessary below for class

Batter. Note: Let's not worry about pitchers for right now and their stats. We'll either do this in class or it will be a part of another programming assignment.

Class Batter attributes

StringBuilders for first name, last name, and position.
Integers (int) for at-bats, hits, walks, doubles, triples, homers, sac flies, strikeouts

Class Batter methods

Default constructor
Constructor that takes in arguments of three StringBuilders, corresponding to first name, last name, and position.
Getters and setters for each of the above attributes.
Batting average – which is hits divided by at-bats. Returned as a double.
On-base percentage – which is (hits + walks)/(at-bats+walks). Returned as a double.
Slugging percentage – ((singles)+(2*doubles)+(3*triples)+(4*homers))/at-bats; note than theoretically, this could top out at 4.0, if a player hits nothing but homers. Returned as a double.
Methods that increment each of the above integer attributes.
Print method that prints the following values for a Batter object, all on one line:

- First name
- Last name
- Position
- At-Bats
- Hits
- Doubles
- Triples
- Homers
- Walks
- Strikeouts
- Batting Average
- On-Base Percentage
- Slugging Percentage

Once you have your class working, then you'll need to insert objects into your code from Assignment 1. You need to pick a collection – either array, ArrayList, or something similar.
Once you get to the ninth batter in the order, the first batter in the lineup bats again.
Add an at-bat for every plate appearance (almost; see next section below), and update the appropriate stats as the result of the play.

Warnings/Weird Scoring Rules That May Not Make Sense

The only time you don't increment a batter's at-bat is when they either walk, or hit a sac fly. If the sac fly scores a run, that batter also receives credit for having batted in a run (keep the second half of this nugget in your mind for a future assignment).

Other Stuff

You'll need to create a method outside of class Batter that prints the column headings. Use the following notations:

- First Name
- Last Name
- pos – for position
- ab – for at-bats
- h – hits
- 2b – doubles
- 3b – triples
- hr – homers
- bb – walks (it's referred to as "bases on balls" in box scores…I don't know why)
- k – strikeouts (might also see referred to on box scores as "so")
- sf – sac flies
- avg – batting average
- slug – slugging percentage
- obp – on-base percentage

Further, you'll need to use printf to align the columns with the outputs from print. Print decimal/double values out to three decimal places.

Print out the box scores/stat lines for each team once the game is finished.

Program output

Print out the result of each play. If runs score, include how many for the given play. Have one play displayed per line. You might choose to do something like this for each batter:

```
Top of the 1st inning.

Up to bat, Fernando Tatis, Jr....
Type 1 to see batter's stats, or 2 to go on to the play: 1
First Name          Last Name           Pos    ab  h 2b 3b hr bb  k sf    avg      slug     obp
Fernando            Tatis, Jr.          ss      0  0  0  0  0  0  0  0    0.000    0.000    0.000
Tatis, Jr.: Pop out.
1 out.
Up to bat, Ryne Sandberg...
Type 1 to see batter's stats, or 2 to go on to the play: 2
Sandberg: Walk!
1 out.
Up to bat, Bryce Harper...
```

Additionally, you'll need to print out the stats of each batter on each team, by batting order.

Example:

```
End of the game.  Final stats:
Team 1
First Name          Last Name           Pos   ab  h  2b  3b  hr  bb  k  sf    avg      slug     obp
Fernando            Tatis, Jr.          ss     3  1   1   0   0   1   1  1    0.333    0.667    0.500
Ryne                Sandberg            2b     4  2   0   0   0   1   1  0    0.500    0.500    0.600
Bryce               Harper              lf     5  1   1   0   0   0   1  0    0.200    0.400    0.200
Roberto             Clemente            cf     4  1   0   0   1   1   0  0    0.250    1.000    0.400
Tony                Gwynn               rf     5  1   0   0   1   0   2  0    0.200    0.800    0.200
Nolan               Arenado             3b     3  1   0   1   0   1   1  1    0.333    1.000    0.500
Albert              Pujols              1b     3  1   0   0   1   0   1  1    0.333    1.333    0.333
Johnny              Bench               c      4  1   0   0   0   0   0  0    0.250    0.250    0.250
Hank                Aaron               dh     3  1   0   0   0   1   0  0    0.333    0.333    0.500


Team2
First Name          Last Name           Pos   ab  h  2b  3b  hr  bb  k  sf    avg      slug     obp
Rickey              Henderson           cf     6  3   1   0   1   1   1  0    0.500    1.167    0.571
Mike                Trout               rf     6  2   1   0   1   0   1  0    0.333    1.000    0.333
Adrian              Beltre              3b     6  3   0   1   2   0   0  0    0.500    1.833    0.500
David               Ortiz               dh     5  3   1   1   0   1   1  0    0.600    1.200    0.667
Frank               Thomas              1b     6  3   1   1   0   0   1  0    0.500    1.000    0.500
Cal                 Ripken, Jr.         ss     5  2   1   0   1   1   2  0    0.400    1.200    0.500
Craig               Biggio              2b     4  1   0   0   0   1   0  1    0.250    0.250    0.400
Ken                 Griffey, Jr.        lf     3  0   0   0   0   3   0  0    0.000    0.000    0.500
Mike                Napoli              c      4  2   0   1   0   2   2  0    0.500    1.000    0.667
```

Your results can – and should – differ from mine, since this has all been randomized.

Submission

Submit your .java files via zip file by 11:55pm on Wednesday, September 9. If you feel it necessary, you may utilize the files we've worked on in class.