**CSC 2220 – Programming in Java**
**Fall 2020 Semester, Block A**
**Programming Assignment 1**
**Due Date: Friday, September 4, 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

We've been playing around with coding up a wooden baseball game during the first week of class. In doing so, we've been able to create classes and methods that determine plays by the roll of two dice. We have already tackled the capability of keeping track of the number of hits in the game, as well as outs.

Now, you'll need to be able to keep score. Typically what you see on the scoreboard during a game are runs, hits, and errors. We're going to replace errors with walks as the third item to show.

## What To Do/Rules of the Game

### Hits and Walks

Now you'll need to keep track of runners on the basepaths. If someone gets a single, they end up on first base. A double – the batter goes to second; triple – the batter goes to third; and a home run means the batter "rounds the bases" and scores a run.

In the situation that a batter hits a single, anyone advances one base. So, if you have a runner on first, they move to second. If there's a runner at second, they move to third. If there's a runner at third, they come home and score a run.

If a batter hits a double, then if there's a runner on first, they move to third. If you have runners at second and/or third and the double is hit, those runners each come home and score.

If a batter hits a triple, if anyone is on base, they'll come home and score. Same situation for anyone on base if a home run is hit.

A walk doesn't advance any runners unless a base is occupied. If a walk is issued, and first base is vacant, then the hitter goes to first and that's it. If a runner is already on first when a walk is issued, then that runner moves onto second, to make room for the current batter. If there's a situation where the bases are loaded (every base is occupied), the person on third comes home and scores a run, the other two baserunners advance, and the batter takes first.

Tally up the number of walks.

Keep up with the number of runs scored in each half inning. You might utilize arrays for this. Or ArrayLists.

### Outs

A ground out results in the batter being out at first, but the other runners advance. It is possible a run could score.

A pop out and a fly out result in the batter being out and no one advancing.

A sac fly means the batter is out, and the runners can advance a base, if the out isn't the third out. A runner could come home from third and score a run. If a runner is on second, they can move to third. If a runner is on first, they can move to second. If no one is on base, it's scored just like a fly out. If the sac fly results in the third out, it's also scored as a flyout.

A double play results in two outs, unless the bases are empty (no base runners). In this situation, you only have one out – the batter. Otherwise, take out the lead runners – if a runner at third, that's an out, then check second, then check first.

## 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.

Find some way to separate half innings. You might have extra carriage returns/new lines to separate each half-inning.

When the game is over (let's assume there can be ties) after nine innings, display the runs scored for each half inning by the away team and the home team. The home team can get at-bats in the bottom of the ninth for right now, even if they're already ahead. Then display total runs, total hits, and total walks for both teams, in a manner similar to the following:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | R | H | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ATL | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | **1** | **5** | **0** |
| PHI | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | - | **4** | **7** | **0** |

Again, remember in lieu of the E column, you'll have a walks column (let's call it W).

# Submission

**Submit your .java files via zip file by 11:55pm on Friday, September 4.** If you feel it necessary, you may utilize the files we've worked on in class.