

CSC 2220 – Programming in Java

Fall 2020 Semester, Block A

Programming Assignment 6-7

Due Date: Thursday, October 8, by 11:55pm, to Moodle

General Rules of Engagement

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.

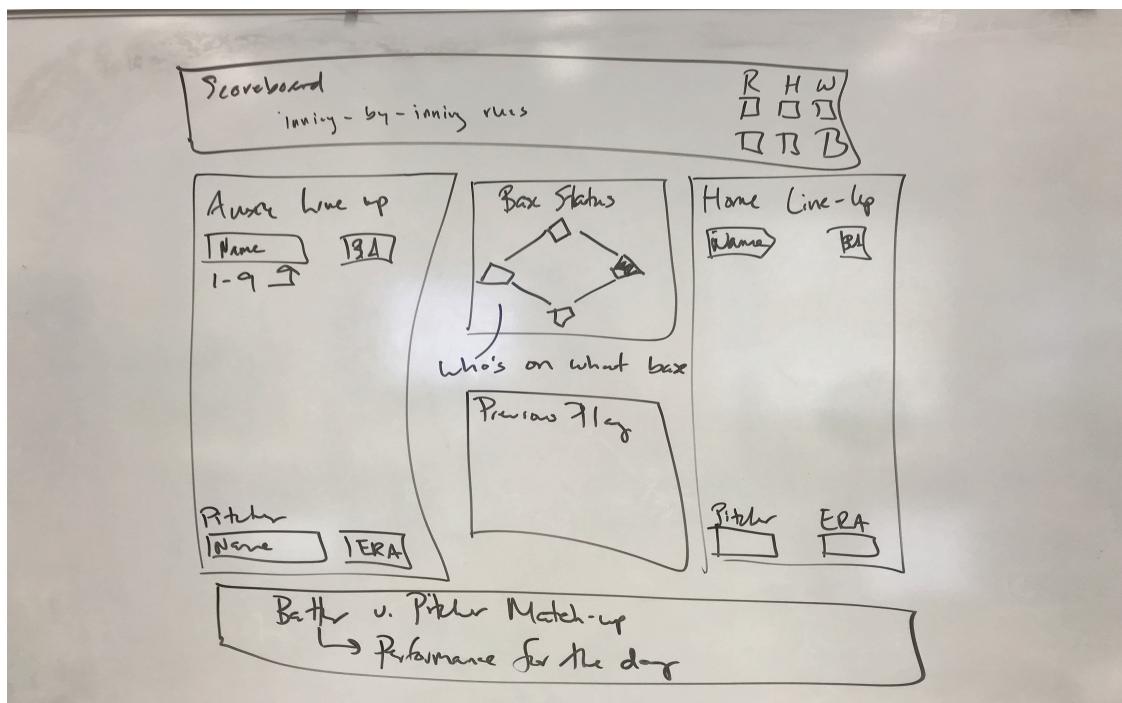
Although you won't necessarily see a deduction, do take care to make sure spelling and spacing have been addressed and are correct.

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



All block long we have been working on various phases of implementing a digital version of a wooden baseball game based on dice rolls. We are now going to implement the same game but with a GUI, as opposed to having to go through command-line to have any fun.

The picture above gives you a good feel for the layout (and what we mean by “low-tech” or “low-fi” prototyping – it was done within minutes on the board). Construct your GUI in a similar fashion. You are all working together to complete this particular programming assignment.

The GUI set-up via the low-tech prototype picture from above:

- Top box: the typical scoreboard you’d see at a baseball game. It lists the inning number, as well as runs scored per inning. Sometimes you see this implemented in groups of three innings, with spaces in between the third and fourth and fourth and seventh innings. Usually ten innings are displayed, as well. To the far right, let’s put total runs, hits, and walks.
- Far left and far right boxes: the batting line-ups for each team, numbered 1-9, along with their position. There’s a separate box for batting average. The last person to be listed, a little further down, is the pitcher’s name and his earned-run average (ERA).
- Top middle box: shows you who’s on what base.
- Bottom middle box: shows you what the previous play was
- Bottom box: shows you the current pitcher v. batter match-up. You should also display how that batter (like maybe they’ve gone 2 for 4 with a single, triple, and sac fly) has fared for today, beyond the first time through the line-up (as batters will all be 0 for 0 against the pitcher).

Miscellaneous:

- You’ll need a way to denote if we’re in the top or bottom of an inning, as well as how many outs there are currently.
- You’ll need a button that’ll start the game. Where you put this button is up to you, but should be labeled and positioned in an obvious location so where the user knows this starts the game. This button should disappear and everything set up (like what we saw in class to some extent on 9/28).
- You’ll need another button for the next play.

Reminders:

- First and foremost, it’s highly recommended that you use the absolute layout in conjunction with Eclipse to do this project. This will save you a lot of time and stress.
- Use lots of panels where you can. You can then set the layout to these to the absolute layout manager, as well. You might find this particularly useful when trying to consistently place text and numbers, as well as boxes for names.

- If you feel that something is either missing or better served elsewhere, let me know. I'm not particular on where the buttons go, necessarily.

Submission

Submit your .java files via zip file by 11:55pm on Thursday, October 8. Only one submission is needed for the group of five.