CMPS 12A
Introduction to Programming
Programming Assignment 2

In this assignment you will write a java program that plays an interactive guessing game with the user. Your program will generate a random integer in the range 1 to 10, then allow the user three guesses to determine the integer. After each guess your program will inform the user whether the guess was correct, too high, or too low. Your source file for this project will be called `Guess.java`, and will define a class called `Guess`. Here is a transcript of several plays of the game. As usual `%` stands for the unix prompt.

```
% java Guess

I'm thinking of an integer in the range 1 to 10.  You have three guesses.

Enter your first guess: 8
Your guess is too high.

Enter your second guess: 5
Your guess is too high.

Enter your third guess: 3
Your guess is too high.

You lose.  The number was 1.

% java Guess

I'm thinking of an integer in the range 1 to 10.  You have three guesses.

Enter your first guess: 5
Your guess is too low.

Enter your second guess: 8
Your guess is too low.

Enter your third guess: 9
You win!

% java Guess

I'm thinking of an integer in the range 1 to 10.  You have three guesses.

Enter your first guess: 5
Your guess is too high.

Enter your second guess: 2
You win!

% java Guess

I'm thinking of an integer in the range 1 to 10.  You have three guesses.

Enter your first guess: 8
You win!
```

Observe that the output includes blank lines at the beginning and end of the program output, and blank lines separating guesses. Your program should match this format exactly.

Use the method `Math.random()` to generate the random integer. Note that this function takes no arguments and returns a random `double` value in the range [0, 1). It is up to you to figure out how to turn that value into a random integer in the set {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Once the mystery number is chosen, program operation will depend on user input. Your program must give truthful responses to the user's guesses, then halt when either the right number is guessed, or the user runs out of guesses. You will use the conditional operations `if` or `if-else` to accomplish this. These commands will be discussed at length in class, and can also be found in Chapter 3 of the text.

What to turn in

Submit your source `Guess.java` to the assignment name `pa2` before the due date. As always start early and ask questions in lab sessions, office hours, and on Piazza.

Discussion

Although this game is very simple, it is interesting to consider what strategy the user might adopt. Try to verify for yourself that if we were to allow four guesses instead of three, then a strategy exists that would find the mystery number with certainty, i.e. the guesser could always win. Similarly if we were to restrict the number to the set {1, 2, 3, 4, 5, 6, 7}, then it can always be found in three guesses. The game, as defined above, does posses an optimal strategy (optimal in the sense that it maximizes the probability winning.) Try to deduce such a strategy, and show that by using it, the guesser will win 70% of the time. As an exercise, write a program that plays the other side of the game (i.e. the role of guesser) and that implements this optimal strategy. This exercise will yield no credit for the current assignment, but it may form the basis of some future project.