

EECS 1021 – Lab D: Conditionals and Loops a.k.a. The Guessing Game

Dr. James Andrew Smith, PEng and Richard Robinson

Summary: In this lab, you will use loops and conditional statements.

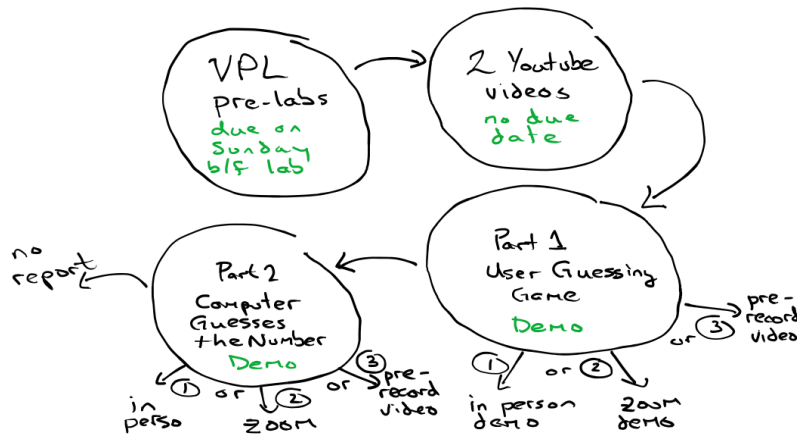


Figure 1 Lab D has two parts. Make sure to do the pre-lab.

Intro

In this lab, you will be creating a basic Java "guessing game" program. This program will use simple, common control flow structures like 'if' statements and 'while' loops.

The goal of the game is to try to guess a randomly generated number, with feedback indicating if the guess is higher or lower than the answer.

The second part of the lab is the same game, except now, the Java program itself will be trying to guess the correct value (in a strategic, non-trivial manner).

Due dates.

- **Pre-Lab:** All of the interactive pre-lab activities are due on the Sunday night before the first of the two lab sessions.
- **Lab Demo:**
 - **Option 1:** live lab demo to the TA (Zoom or in-person)
 - **Option 2:** record a screen capture and submit a video to eClass.

Marking Guide:

- All interactive Pre-lab activities are graded out of 1 and count towards your "interactive" activity grade. Any other pre-lab activity is not graded.
- Part 1: 0.6 marks. 0.3 if partially successful. 0 if not attempted.
- Part 2: 0.4 marks. 0.2 if partially successful. 0 if not attempted.

Pre-lab

Check for pre-lab activities in Module 5. These could be either H5P activities or VPL activities or both. All pre-lab activities are due on the Sunday before the lab at 11:55pm.

Specification

- The generated random value should be in the range (0, GuessingGame.UPPER_BOUND].
 - From 0 to the “upper bound”.
 - Make the “upper bound” equal to 100 to begin with
 - For your demonstration, make the upper bound the value of the day of your demonstration (e.g. 10 if it's February 10). If you are recording the demo, then set it to the due date.
- Before each guess, the standard output should print Enter a guess: (Make sure that it does not print a newline character after this).
 - Hint: print() vs. println()
- After the user enters a guess:
 - If the guess is correct, print "Correct! Took ? guesses" where ? is the number of guesses (hint: a counter!)
 - i. The program should then terminate.
 - If the guess is greater than the correct value, print "Guess is higher than answer\n". Likewise, for if the guess is less than the correct value.
- All the code should be within the main method of the GuessingGame class

Tips

- The most important tip -- if you don't know how to do something, always try to Google it first. This is about 99.9% of what programming is.
 - GeeksForGeeks Java page: <https://www.geeksforgeeks.org/java/>
 - Kishori Sharan's book @ the YorkU Library (<https://bit.ly/3stXeWT>)
 - The Java for Engineers book @ the YorkU Library (<https://bit.ly/3rBvO2o>)
 - LinkedIn Learning's "Learning Java" videos. (<https://www.linkedin.com/learning/learning-java-4/>)
- In most sources online, you will see a variable declaration like int i = 0. However, with modern Java, you can just use var, like var i = 0. This approach is recommended.

```
[jshell] while(scanObject.hasNextInt()){
...>     var myGuess = scanObject.nextInt();
...>     System.out.println(myGuess);
...> }
1
1
2
2
10
10
1000
1000
jjjjjjjj ← exit loop b/c hasNextInt is false.
```

Figure 2 Watch the YouTube video:
<https://youtu.be/gidNON1d0QU>

```
package eeecs1021;

import java.util.Random;
import java.util.Scanner;

/**
 * Part A: A simple guessing game. Keep guessing numbers until you get the correct answer. The goal of the game
 * is to complete it using as little guesses as possible.
 */
public class Part1 {

    public static void main(String[] args) {
        var RandomNumber = new Random(); // random object

        var scanObject = new Scanner(System.in);

        while () {

            if () {

            }

            System.out.println("Correct! Took " + counter + " guesses");

        }
    }
}
```

Skeleton for Part1.java (with lots of stuff missing) ... what should this class really be called?

Procedure

1. Create a random number generator using the Random class.
2. Create a random number in the specified range.
➤ <https://youtu.be/sMgbD3ISA6g>
3. Use the Scanner class to read input values.
➤ Don't know how to use Scanner? Search it online and/or read the Java documentation for it)
➤ <https://youtu.be/gidNON1d0QU>
4. Use a while loop to repeat actions for each guess.
5. Within the loop, check if the guess is
➤ == (equal)
➤ > (greater than)
➤ < (less than)
the answer, and take appropriate actions.

```
import java.util.Random;
" " " " Scanner;
```

```
public class NameTheClass {
    Make a constant UPPER_BOUND
```

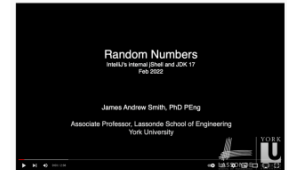
```
    a main method {
```

```
        Make a Random object
```

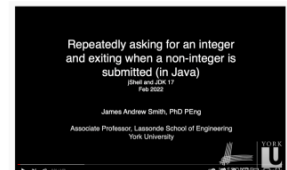
```
        put a value into that object
        that doesn't exceed
        UPPER_BOUND
```

```
        Make a Scanner object
        that accepts System.in
```

```
        while loop
        {
            REPEAT (check hasNextInt())
            {
                → check hasNextInt()
                → assign scanned input
                  to a variable
                → break if value is right
                → tell user if it's
                  too low or too high
            }
        }
    }
```



<https://youtu.be/sMgbD3ISA6g>



<https://youtu.be/gidNON1d0QU>

YouTube videos: <https://youtu.be/sMgbD3ISA6g> and
<https://youtu.be/gidNON1d0QU>

Demonstrate Part 1 for marks.

You can do the **demo of Part 1** in one of three ways:

1. In person in the William Small Labs during your scheduled lab time.
2. Over Zoom during your scheduled lab time.
3. As a pre-recorded video (1 minute long – not longer!) uploaded to eClass on the Sunday after the lab.

Note that you need to set your upper bound to match the date. Make the “upper bound” equal to 100 to begin with. For your demonstration, **make the upper bound the value of the day** of your demonstration (e.g. 10 if it's February 10). If you are recording the demo, then set it to the due date (i.e. the numeric day of the Sunday that you hand it in).

Part 2 [0.4 marks]: The Computer Does the Guessing Game

Like Part 1, but this time, don't have the user make the guesses. Have your program do it.

Specification

1. The generated random value should be in the range $(0, \text{GuessingGame.UPPER_BOUND}]$.
 - a. The range includes zero but does not include the upper bound (it's one value less than the upper bound).
2. After each guess, the standard output should print "Guess: 42" (where 42 is the guessed value)
3. After each guess:
 - a. If the guess is correct, print "Correct" and exit the program.
 - b. Otherwise, print "guess > answer" or "guess < answer"

Additionally, the guessing algorithm should work in the same way that a human would guess. That is, instead of repeatedly guessing a number in $(0, 100]$ each time, the lower or upper bounds are modified after each iteration to limit the range so that fewer guesses are needed. For example:

Iteration	Guess	Answer	Range
0	2	50	(0,100)
1	75	50	(2, 100)
2	40	50	(2, 75)
3	60	50	(40, 75)
4	50	50	(40, 60)

Procedure

1. Create a random number generator using the `Random` class.
 - a. This means make ("instantiate") a `Random` object.
2. Create a random number in the specified range from that `Random` object.
 - a. Include both upper and lower bounds.
3. Using an infinite `while` loop, have your program guess a random number in each iteration. The random number should be generated bounded to the most restrictive limits (as described in the Specification section).
 - a. Again, this is different than in Part 1. In Part 1, the user was doing the guessing using the scanning method. In Part 2, the Java program is to do the guessing. Do the guess by using the `nextInt()` method on the `Random` object, while inside the `while` loop.
4. After each guess, check if it is `==` or `>` or `<` the answer, and take appropriate actions.
 - a. Only exit if it is `==`.

Hint: that the `Random` class has a method to generate a random `int` with an upper bound, but not a lower bound. To circumvent this, some math is needed to adjust the value:

```
MyRandomObject.nextInt(BigValue - SmallValue) + SmallValue;
```

Demonstrate Part 2 for marks.

You can do the **demo of Part 2** in one of three ways:

1. In person in the William Small Labs during your scheduled lab time.
2. Over Zoom during your scheduled lab time.
3. As a pre-recorded video (1 minute long – not longer!) uploaded to eClass on the Sunday after the lab.

Note that you need to set your upper bound to match the date. Make the “upper bound” equal to 100 to begin with. For your demonstration, **make the upper bound TWICE value of the day** of your demonstration (e.g. **20** if it's February 10). If you are recording the demo, then set it to the due date (i.e. the numeric day of the Sunday that you hand it in).