

Lab 02 Specification – Java Programming Review
Due (via Bitbucket and hard copy) Wednesday, 13 September 2017
50 points

Lab Goals

- Refresh your ability to write good code in Java
- Write two versions of the same program with different behaviors
- Think about how to solve a variety of programmatic challenges

Assignment Details

Now that we have written some Java code together in the last few lectures, it is your turn. In this lab, you will practice writing two different versions of the “Guess My Number” game. In the first, the computer will select a number and the user will attempt to guess it correctly. In the second, the user will select a number and the computer will attempt to guess it correctly.

In addition to these two programs, you will also be presented with three algorithms of different complexities. While you do not need to provide code solutions, you should carefully think about how to solve each of these challenges.

Guess My Number, Human Player (20 points)

Write a Java program that implements the “Guess My Number” game. The program should choose a number between 1 and 100 inclusive, and prompt the user to guess that number. After each guess, the system should print whether the guess was too high, too low, or correct. If correct, the program should print the number of guesses that the user made, and then prompt the user to play again.

Sample run:

```
aldenv27:lab2 amohan$ javac lab2part1.java
aldenv27:lab2 amohan$ java lab2part1

Hello user!  I'm thinking of a number between 1 and 100.
See if you can guess it!
Your guess: 50
Your guess was too high!
Your guess: 45
Your guess was too high!
Your guess: 40
Your guess was too low!
Your guess: 42
You got it in 4 tries!  My number was 42.

Would you like to play again (y/n)? y
```

```
Hello user! I'm thinking of a number between 1 and 100.
See if you can guess it!
Your guess: 19
Your guess was too high!
Your guess: 7
You got it in 2 tries! My number was 7.

Would you like to play again (y/n)? n
```

Guess My Number, Computer Player (20 points)

In the previous section, you wrote a Java program that plays the “Guess My Number” game. Now, you will modify that program to switch roles:

- YOU will choose the number.
- The COMPUTER will attempt to guess the number.
- On each iteration, YOU must inform the computer whether its guess is too high, too low, or correct.
- The COMPUTER will repeat your feedback (useful for debugging).
- When guessing correctly, the COMPUTER will report how many guesses it took, and then ask if you want it to guess again.

When giving feedback to the computer about its guess, you should use the following:

- 1 = The computer's guess is too high.
- 0 = The computer's guess is correct.
- -1 = The computer's guess is too low.

Your input prompts and output labels should look similar to the ones in this sample run:

```
aldenv27:lab2 amohan$ javac lab2part2.java
aldenv27:lab2 amohan$ java lab2part2

Hello user! Please pick a number between 1 and 100.
My first guess is 50. How did I do?
1

Hmmm, my guess was too high. My next guess is 25. How did I do?
-1

OK, my guess was too low. My next guess is 37. How did I do?
1

Hmmm, my guess was too high. My next guess is 31. How did I do?
1

Hmmm, my guess was too high. My next guess is 28. How did I do?
0
```

```
Yay, I got it in 5 tries! Your number was 28.
Would you like me to guess again (y/n)? y

Hello user! Please pick a number between 1 and 100.
My first guess is 50. How did I do?
1

Hmmm, my guess was too high. My next guess is 25. How did I do?
0

Yay, I got it in 2 tries! Your number was 25.
Would you like me to guess again (y/n)? n
```

There are three ways that you can have the computer guess your number. In the first case, the computer can simply try every number between 1 and 100 in order. In the second case, you can just pick random numbers between 1 and 100 as guesses. Each of these will give you partial credit. In the third case, you can keep track of the largest and smallest number that it could possibly be, and always guess the number in the middle. This will give you full credit.

Note that I do not expect your program to detect when you are lying to the computer about your number. If you want to feel superior to the machine, be my guest, but make sure that your program will respond to “real” input as well.

Additional Questions (10 points)

Please also take the time to answer the following questions:

1. Describe at an algorithmic level (no code required) how a Java method can count the number of vowels in a provided `String`.
2. Describe at an algorithmic level how a Java method can take a provided integer n and return the sum of all positive, odd integers less than n (with input 13, the return value should be $1 + 3 + 5 + 7 + 9 + 11 = 36$).
3. Describe at an algorithmic level how a Java method can take an array of integers and determine whether or not they are all distinct ($\{1, 2, 3, 4, 5\}$ is all distinct; $\{1, 2, 3, 4, 2\}$ is not).

Submission Details

For this assignment, please submit the following to your `cs112s2017-<your user name>` repository (and ensure that the instructor has access to your repository):

1. Commented source code from the “Guess My Number, Human Player” program.
2. Commented source code from the “Guess My Number, Computer Player” program.
3. A document containing responses to the “Additional Questions” prompts.