

CSCE 145: Lab 10

A Guessing Game!

Objective: In doing this lab you will learn about iteration (one of the three basic control mechanisms in programming, the others being sequence and branching). You will also develop a command-line interface for your software.

Specification: Use Java to write and run a guessing game program that will think of a number and invite a user to guess it. After each guess, the game will tell the user if his/her guess was high or low or if they guessed correctly, and then ask the user to guess again. The game will continue to ask for the next guess until either the user guesses correctly or gives up. When the game ends, if the user finally guessed the correct number, the game will tell the user how many guesses were needed and whether the user was a good guesser or not. If the user quits before guessing correctly, he/she will be suitably insulted.

Here are two examples of how the game should work:

```
Welcome to the Guessing Game.
I am thinking of a number between 0 and 1000.
```

```
What is your guess? 25
Good guess, but 25 is too low.
```

```
What is your guess? 768
Good guess, but 768 is too high.
```

```
What is your guess? 432
Correct! It took you only 3 guesses. You win!!
```

```
Or,
Correct, it took you 23 guesses. Don't waste your money on the lottery and stay away from quiz shows.
```

```
Welcome to the Guessing Game.
I am thinking of a number between 0 and 1000.
```

```
What is your guess? 222
Good guess, but 222 is too low.
```

```
What is your guess? -55
Quitting so soon? Loser!
```

Note that a negative guess indicates that the user wants to quit.

Your program will have to use a random number generator to enable it to think of a different number each time it is run. Fortunately, there is one built into Java. Here is how you would invoke it:

```
int secretNumber;
secretNumber = (int)(Math.random() * 1000. + 1);
```

`Math.random()` generates a random double between 0.0 and 0.99999999. When it is multiplied by 1000 and then 1 is added to it, it has a value between 1.0 and 1000.999999. Casting it to an int produces an integer from 1 to 1000.

Once your game has its secret number, it begins asking a user to guess it until correct or he/she quits. Use a loop for this part, and I suggest a while-loop controlled by two Boolean variables: one (stop) that indicates when the user wants to quit and the other (correct) that indicates when the guess is correct, as follows:

```
while (!stop && !correct) {
```

Please use `JOptionPane` for your program's inputs and outputs.

As always, the source code must be indented properly, have a header with your name and a short description of the purpose of the program, and contain comments that describe important parts of the algorithm.

Upload your program to the CSE Dropbox, located at dropbox.cse.sc.edu.