

CSE 190 D, Winter 2013

Programming Assignment #3: Guessing Game (15 points)

Due: Tuesday, January 29, 2012, 11:30 PM

This assignment focuses on `while` loops, `if/else` and random numbers. Turn in a file named `GuessingGame.java`.

Your program allows the user to play a game in which the program thinks of a random integer and accepts guesses from the user until the user guesses the number correctly. After each incorrect guess, you will tell the user whether the correct answer is higher or lower. Your program must exactly reproduce the format and behavior of the logs in this document.

The log below shows one sample execution of your program. Your output will differ depending on the random numbers chosen and user input typed, but the overall output structure should match that shown below.

```
I'm thinking of a number between 1 and 100...
Your guess? 50
It's lower.
Your guess? 25
It's higher.
Your guess? 35
It's lower.
Your guess? 30
It's higher.
Your guess? 32
It's lower.
Your guess? 31
You got it right in 6 guesses!
Play again (1=yes 2=no)? 1

I'm thinking of a number between 1 and 100...
Your guess? 50
It's higher.
Your guess? 75
It's lower.
Your guess? 65
It's lower.
Your guess? 64
You got it right in 4 guesses!
Play again (1=yes 2=no)? 1

I'm thinking of a number between 1 and 100...
Your guess? 60
It's lower.
Your guess? 20
It's higher.
Your guess? 30
It's higher.
Your guess? 40
It's higher.
Your guess? 50
It's lower.
Your guess? 47
It's higher.
Your guess? 49
You got it right in 7 guesses!
Play again (1=yes 2=no)? 2

Overall results:
Total games = 3
Total guesses = 17
```

First, a series of guessing games is played. In each game, the computer chooses a random number between 1 and 100 inclusive. The game asks the user for guesses until the correct number is guessed. After each incorrect guess, the program gives a clue about whether the correct number is higher or lower than the guess. Once the user types the correct number, the game ends and the program reports how many guesses were needed.

After each game ends and the number of guesses is shown, the program asks the user if he/she would like to play again. Assume that the user will type an `int` as the response to this question.

A new game should begin if the user's response is 1. If the user types 2 that means the user does not want to play again. You do not need to handle any other input values.

Once the user chooses not to play again, the program prints overall statistics about all games. The total number of games and the total guesses made in all games are displayed.

Your statistics should be correct for any number of games or guesses ≥ 1 .

You should handle the special case where the user guesses the correct number on the first try. Print a message as follows:

```
I'm thinking of a number between 1 and
100...
Your guess? 71

You got it right in 1 guess!
```

Assume valid user input. When prompted for numbers, the user will type integers only, and they will be in proper ranges.

Implementation Guidelines:

Produce randomness using a single `Random` object, as seen in lecture. Remember to `import java.util.*;`

Read user answers using the `Scanner`'s `nextInt` method (not `nextLine`, which can cause strange bugs when mixed with `nextInt`). If you get an `InputMismatchException`, you are trying to read the wrong type of value from a `Scanner`.

Produce repetition using `while` or `do/while` loops. Some students try to avoid properly using `while` loops by writing a method that calls itself, or a pair of methods A and B where A calls B and B calls A, creating a cycle of calls. Such solutions are not appropriate on this assignment and will result in a deduction.

```
I'm thinking of a number between 1 and 100...
*** HINT: The answer is 46
Your guess? 50
It's lower.
Your guess? 25
It's higher.
Your guess? 48
It's lower.
Your guess? 46
You got it right in 4 guesses!
```

(suggested initial simple version of program)

We suggest that you begin by writing a simpler version that plays a single guessing game. Ignore other features such as multiple games and displaying overall statistics.

While debugging it is useful to print a temporary "hint" message like that shown at left. This way you will know the correct answer and can test whether the program gives proper clues for each guess. This is also helpful for testing the "1 guess" case.

Style Guidelines:

For this assignment you are limited to the language features shown in lecture.

Use whitespace and indentation properly. Limit lines to 100 characters. Give meaningful names to variables, and follow Java's naming standards. Localize variables. Put descriptive comments at the start of your program and on any complex sections of code.