# TCSS143 Fundamentals of Object-Oriented Programming
# Theory and Application
# Programming Assignment #5
# 15 Points

## Description:

This assignment will give you practice with while loops and pseudorandom numbers. You are going to write a program that allows the user to play a simple guessing game in which your program thinks up an integer and allows the user to make guesses until the user gets it right. For each incorrect guess you will tell the user whether the right answer is higher or lower. Your program is required to *exactly* reproduce the format and behavior of the log of execution at the end of this write-up, so you may want to look that over first.

At a minimum, your program should have the following static methods in addition to method main:

- a method to give instructions to the user
- a method to play one game with the user (just one game, not multiple games)
- a method to report overall results to the user

You may define more methods than this if you find it helpful, although you will find that the limitation that methods can return only one value will tend to limit how much you can decompose this problem.

You are to define a class constant for the maximum number used in the guessing game. The sample log shows the user making guesses from 1 to 100, but the choice of 100 is arbitrary. By introducing a constant for 100, you should be able to change just the value of the constant to make the program play the game with a range of 1 to 50 or a range of 1 to 250 or some other range starting with 1.

When you ask the user whether or not to play again, you should use the "next()" method of the Scanner class to read a one-word answer from the user. You should continue playing if this answer begins with the letter "y" or the letter "Y". Notice that the user is allowed to type words like "yes". You are to look just at the first letter of the user's response and see whether it begins with a "y" or "n" (either capitalized or not) to determine whether to play again.

Assume that the user always types an integer when guessing, that the integer is always in an appropriate range and that the user gives you a one-word answer beginning with "y", "Y", "n" or "N" when asked whether to play again.

You will notice at the end of the log that you are to report various statistics about the series of games played by the user. You are to report the total number of games played, the total number of

guesses made (all games included), the average number of guesses per game and the maximum number of guesses used in any single game.

Here are a few helpful hints to keep in mind.

- This program needs to generate pseudorandom numbers.

- To deal with the yes/no response from the user, you might want to use some of the String class methods. You will want to use the next() method of the Scanner class to read a word from the console.

- Because this program uses pseudorandom numbers, you won't be able to recreate this exact log. The key requirement is that you reproduce the *format* of this log and that your calculations for overall statistics are correct for your log.

- It's a good idea to change the value of your class constant and run the program to make sure that everything works right with the new value of the constant. For example, turn it into a guessing game for numbers between 1 and 5.

In the last program I asked you to write very short methods that were no longer than 15 lines long and to have a very short main. This program is more difficult to decompose into methods, so you may end up having methods that are longer than 15 lines. You can also include more code in your main method than I allowed in the last program. In particular, you are required to have a while loop in main that plays multiple games and prompts the user for whether or not to play another game. You shouldn't have all of the code in main because you are required to have the methods described at the beginning of this write-up.

I will once again expect you to use good programming style and to include useful comments throughout your program. I will expect you to make appropriate choices about when to store values as int versus double, which if/else constructs to use, what parameters to pass, and so on.

Your program should be stored in a file called **Guess.java** and submit on angel. Points will be taken off if you submit in any other format.

## Log of execution (user input underlined)

```
This program allows you to play a guessing game.
I will think of a number between 1 and 100
and will allow you to guess until you get it.
For each guess, I will tell you whether the
right answer is higher or lower than your guess.

I'm thinking of a number...
Your guess? 20
higher
Your guess? 40
higher
Your guess? 60
higher
Your guess? 80
```

```
higher
Your guess? 100
lower
Your guess? 90
lower
Your guess? 88
lower
Your guess? 86
You got it right in 8 guesses

Do you want to play again? Yes

I'm thinking of a number...
Your guess? 20
higher
Your guess? 40
higher
Your guess? 60
higher
Your guess? 80
higher
Your guess? 82
higher
Your guess? 84
higher
Your guess? 86
higher
Your guess? 88
higher
Your guess? 90
higher
Your guess? 92
higher
Your guess? 94
lower
Your guess? 93
You got it right in 12 guesses

Do you want to play again? YES

I'm thinking of a number...
Your guess? 20
higher
Your guess? 40
higher
Your guess? 60
lower
Your guess? 58
lower
Your guess? 56
You got it right in 5 guesses

Do you want to play again? No

Overall results:
    total games   = 3
    total guesses = 25
```

```
guesses/game  = 8.333333333333334
max guesses   = 12
```

## Submission and Grading:

There will be points taken off for not following the conventions listed in this document regarding submissions, outputs and naming conventions.

You are required to properly indent your code and will lose points if you make significant indentation mistakes. See the coding conventions document on the course web page for an explanation and examples of proper indentation.

Give meaningful names to methods and variables in your code. Localize variables whenever possible -- that is, declare them in the smallest scope in which they are needed.

Include a comment at the beginning of your program with basic information and a description of the program **and include a comment at the start of each method**. Your comments should be written in your own words and not taken directly from this document.

You should include a comment at the beginning of your program with some basic information and a description of the program, as in:

```
// Menaka Abraham
// 1/28/16
// TCSS 143
// Assignment #5
//
// This program will...
```

You should name your file **Guess.java** and you should turn it in electronically on the class web page.