



# COMPUTER SCIENCE 12B (FALL, 2020) ADVANCED PROGRAMMING TECHNIQUES IN JAVA

## PROGRAMMING ASSIGNMENT 2

### Program Description:

This assignment will test your understanding of the use of the `Scanner` objects, `Strings`, `Random` objects, static methods, and everything covered so far.

You should limit yourself to the Java features covered in class so far (lecture 5).

**Modularity in your code is very important, YOU MUST USE STATIC METHODS.**

### Problem 1:

Write a program that allows the user to play a simple guessing game in which you think up an integer and have the computer guess the integer until it gets it right. For each incorrect guess you will tell the program whether the right answer is higher or lower. Your program is required to *exactly* reproduce the format and behavior of the log of execution.

You are to define a class constant for the maximum number used in the guessing game. The sample log shows the program 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 "higher", "lower", or "correct" in response to the program's guess, that the user guesses an appropriate number and answers truthfully, 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.

~~20~~ 101 20 0-99+1 => 1-100  
~~20~~ random a higher a-bw  
 (max. input)

### Log of execution (user input bolded)

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

Think of a number...

My guess: 66

**higher**

My guess: 90

**lower**

My guess: 88

**lower**

My guess: 85

**correct**

I got it right in 4 guesses

Do you want to play again? **Yes**

Think of a number...

My guess: 20

**lower**

My guess: 15

**lower**

My guess: 12

**lower**

My guess: 8

**higher**

My guess: 10

**lower**

My guess: 9

**correct**

I got it right in 6 guesses

Do you want to play again? **No**

Overall results:

total games = 2

total guesses = 10

guesses/game = 5.0

max guesses = 6

Answer 60

Guess 70

lower

Next Guess should < 70

Guess 30

higher

Next Guess should > 30

### **Problem 2:**

Write a reverse Hangman game in which the user thinks of a word and the computer tries to guess the letters in that word. The user tells the computer how many letters the word contains. Your program must output what the computer guessed on each turn and show the partially completed word. It also must use pseudorandom functions to make guesses (i.e., it should not simply try all the letters in order, nor should it use the user's input to its advantage) and it should not guess the same letter more than once.

NOTE:

1. The only String methods that you are allowed to use are: length(), substring(), charAt(), toUpperCase(), toLowerCase().

2. 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.

**Log of execution (user input bolded)**

This program plays a game of reverse hangman.  
You think up a word (by typing it on the computer) and I'll try to guess the letters.

How many letters are in your word? **5**  
Please enter the word for me to guess (letters only): **hello**

- - - - -

```
+---+
|   |
|   |
|   |
|   |
+-----
```

*whe*

I've got 0 of the 5 letters so far  
I guess: M  
Is that letter in the word? **n**

- - - - -

```
+---+
|   |
|   0
|   |
|   |
+-----
```

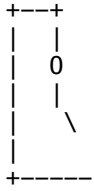
I've got 0 of the 5 letters so far  
I guess: U  
Is that letter in the word? **n**

- - - - -

```
+---+
|   |
|   0
|   |
|   |
+-----
```

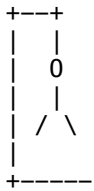
I've got 0 of the 5 letters so far  
I guess: G  
Is that letter in the word? **n**

- - - - -



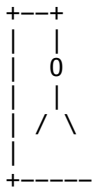
I've got 0 of the 5 letters so far  
I guess: P  
Is that letter in the word? n

- - - - -



I've got 0 of the 5 letters so far  
I guess: E  
Is that letter in the word? y  
How many of that letter are in the word? 1

- e - - -



I've got 1 of the 5 letters so far  
I guess: V  
Is that letter in the word? n

- e - - -



I've got 1 of the 5 letters so far  
I guess: H  
Is that letter in the word? y  
How many of that letter are in the word? 1

h e - - -



```

| 0
| | \
| /  \
+-----

```

I've got 2 of the 5 letters so far

I guess: I

Is that letter in the word? n

h e - - -

```

+---+
|  |
| 0 |
| / | \
| /  \
+-----

```

I've got 2 of the 5 letters so far

I guess: B

Is that letter in the word? n

h e - - -

```

+---+
|  |
| 0 |
| / | \
| /  \
+-----

```

You beat me this time.

### Grading:

You will be graded on

- **External Correctness:** The output of your program should match exactly what is expected. Programs that do not compile will not receive points for external correctness.
- **Internal Correctness:** Your source code should follow the stylistic guidelines linked in LATTE. Also, remember to include the comment header at the beginning of your program.

### Submission:

Create a folder containing your Java source code (programs). Compress (zip) the folder and upload it to Latte by the day it is due. For late policy check the syllabus.