
CMSC 201 Fall 2016

GRS Worksheet – Week 5 – Guessing Numbers

High or Low:

The code on the back of this sheet simulates the game you just played, to guess a number between zero and 100, using only “higher,” “lower,” or “correct” to figure out the right answer. What strategy would you use to win?

Drawing a Blank:

On the back of this worksheet, there was supposed to be Python code that would let you a user play this game on the computer. Unfortunately, data corruption has destroyed all of the most important bits of the code. The variables, function calls (like `main()` and `input()`), and operators have been replaced with blanks (the blank’s length doesn’t mean anything).

Fortunately, the programmer commented their code, and the comments weren’t blanked by the data corruption. You should be able to use the comments to fill out the code so it works correctly again. (It may help to ignore the blanks and simply pseudocode/flowchart your own answer first.)

NOTE: To make the code a bit simpler, the magic number 100 is allowed.

A New Tool:

We haven’t covered libraries in class yet, and we won’t cover them in any detail for quite a while. However, our code below makes use of a library to allow us to play a new guessing game each time.

Libraries are simply code that has already been written for you (like `print()`) that you can *choose* to make use of. We do this by “importing” the library, either in whole or in part. The first line of the code below imports only the `randint()` function from the `random` library.

```
from random import randint
```

Even without having seen it before, what do you think the `randint()` function does? What sort of information do you think it needs to know in order to do exactly what we want? (*Hint: It generates a value for us.*)

```

from random import randint # allows us to randomly generate number to guess

_____():

# get a random number from 0 to 100; randint() includes the ending number
num = randint(_____, _____)

# get the user's guess
guess = _____ ("Guess a number from 0 to 100, inclusive: ")
numGuesses = _____

# keep going until they've guessed the correct number
_____ :

    # tell user what direction to guess in next
    _____ :
        _____ ("higher!")
    _____ :
        _____ ("lower!")

# ask the user for their next guess
_____ = _____ ("What is your next guess (0-100)? ")
_____ = _____ + 1

_____ ("You won! And it only took you", _____, "guesses.")

_____ ()

```