

# CSCI 141 - Spring 2019

## Assignment 3: Conditionals and Loops

Due Date: See Canvas

This assignment has 2 parts. For the first part, answer the questions on Canvas; for the second portion you will complete a single programming task.

**Reminder:** You can discuss this assignment with your peers. However, the answers to the questions and programming solution **MUST** be your own. You cannot copy another person's code, you cannot have another person tell you what code to type, etc. If any part of this is unclear, please come see me.

### 1 Questions: 18 points

Please answer the questions available on Canvas. The questions on Canvas have been configured so that there is no time limit, but you have only 2 attempts to submit your answers. The score that is recorded in Canvas is the score that is the latest (most recent submission) of your attempts.

### 2 Guessing Game: 32 points

Assume you are a computer programmer working for a company, *Nostalgia-R-Us*, that makes legacy (old-style, text-only) games for people who were using computers in the early 1980s. The game you have been tasked to write is a simple guessing game. A player specifies how many tries are allowed, and then proceeds to guess a secret two-character sequence. Because the game is intended for distribution to alumni of Western Washington University, the letters are selected from the letters in the word **bellingham**. See the sample screen shots in Figure 1 for sample gameplay.

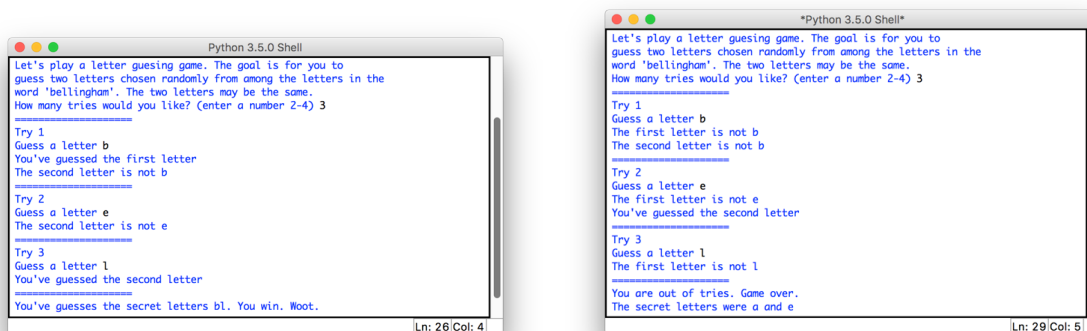


Figure 1: Two sample runs of the program

Your manager has provided you with the following requirements and/or pseudocode:

- Your program must be named `letterGuessGame.py`.
- The program should provide a brief blurb that explains the game and prompts the player to specify the number of tries.
- Two letters from the word `bellingham` should be chosen **randomly** as the secret answer. Because the letters are chosen independently, both the first and second secret letters may be the same.
- While the player has tries remaining, the game should prompt the user to guess a letter. For each of the two secret letters, the program should specify whether the guess was correct or not. If the guess is not one (or neither) of the secret letters, then the program should output a statement stating that fact.
- The program should remember which letter(s) (if any) have already been guessed correctly. Once a letter has been guessed correctly, subsequent output should not mention that letter.
- If the player guesses both letters, the game should output "You win" and terminate right away, even if the player has tries remaining.
- If the player does not correctly guess the secret letters in the number of tries indicated, the game should end, specify that there are no more tries remaining, and the correct answer should be revealed.

This game can be implemented many different ways. Declare and use as many variables as you need to keep track of guesses. The logic for a sample "you lose" game play is shown below.

**Num Guesses** : 4

**Secret Answer** : bh

**User Guess 1** : b

**Game Response** : You have guessed the first letter. The second letter is not b.

**User Guess 2** : g

**Game Response** : The second letter is not g.

**User Guess 3** : l

**Game Response** : The second letter is not l.

**User Guess 4** : e

**Game Response** : You are out of tries. Game over. The secret letters were b and h.

## 2.1 Testing

Test your program thoroughly. Make sure it works correctly for a "win" and a "lose" scenario, noting that a player can win after a minimum of 1 attempt (if both letters are the same and the player guesses them on the first try) but can also win in more than that, up to (and including) the maximum number of attempts specified.

## Submission

Upload `letterGuessGame.py` to Canvas.

## Rubric

Canvas (written) questions	18 points
Top of python file contains comments, including your name	1
The program correctly prompts for the number of tries	2
The program selects two random characters from the letters in <code>bellingham</code>	4
The program correctly keeps track of how many tries are remaining	4
The program specifies which (if any) of the secret letters have been guessed correctly after each guess	4
If one of the two letters has been guessed correctly, on subsequent guesses, the program does not mention the already-guessed letter	5
The program terminates right away and says "Win" if the player guesses correctly. If the player loses, the answer is revealed.	5
The program runs as intended, and does not generate errors	5
The code is commented adequately and variable names are appropriately named	2
Total	50 points

## 3 Challenge Problem

This challenge problem is worth two points of extra credit: Write a program that prompts the user for a non-negative decimal number, then prints the binary representation of the number with no leading zeros.

Submit your Challenge Problem solution in a file named `binary.py` to the A3 Challenge (**NOT** A3 Code) assignment on Canvas.