Programmer Journal / Mini Assignment - Unit 2 Lesson 7

**Mini assignment (Hi Low)**

I had an issue where I could enter a number that wasn’t in range of 1-100 so I added an if statement to make sure if it wasn’t in range it would continue the loop. I had difficulty at the end of the game when I would say I wanted to play again where the counter wouldn’t reset back to zero and would only give one attempt. To fix this I added counter=0 before the break at the end of the guess loop so that it would reset and also after you input an invalid number. I encountered an issue where after we have guessed the number, when we say yes to playing again, it wouldn’t restart the game to ask for another guess. To fix this after the game when the number was guessed, I had to set the guess back to -1, so that it would allow the game to start over. After this fix, the secret number wouldn’t generate a new one so I had to add a line in the same place so that it would regenerate a new number to be put into temporary storage there. After these changes, the game seems to be running perfectly in all of my testing.

# UNIT 2: Mini Programming Project - Number Guessing Game

#

# Purpose to program the childhood game hi-low. The computer picks

# a number between 1-100 and the human player tries to guess the number

# in the fewest number of rounds. The game should keep track of how

# many attempts the player makes guessing the right number.

# Should the player take ten attempts at guessing the game ends displaying

# the correct answer and asks the user if they want to play again.

#

# Author: Cole Becker

#

# Date: September 18, 2023

# Setup the environment

import random # the library 'tool' used to generate pesudo random values

number = random.randint(1,100) # pick a random value between 1 and 100

data = "NA" # the user's data to check valid input of whole numbers

counter = 0 # keep track of the number of guesses

play = False # game flag for main loop and play again feature

guess = -1 # an impossible starting value, human's guess

# Inform the human player about the program - specific to CLI programs

print("Welcome to the guessing game: Hi-Low.")

print("I'm thinking of a number between 1 and 100")

print("You have ten attempts to guess the right number")

print("Would you like to play the game?")

print()

data = input("yes (y/Y) or no (n/N): ")

# check the player's response and set the play flag to True or False

if data.lower() == "y":

play = True

elif data.lower() == "n":

play = False

print("Thanks for playing! bye.")

else:

print("Error, please enter 'y/Y' or 'n/N'.")

# Write a while loop as the main game loop

while play: # your condition here:

# Write the guess and check loop using variables guess and number

while guess != number: # your condition here:

while True:

counter += 1 # add one to the guess counter

data = input("please enter your guess from 1 to 100: ") # ask the user for a number between 1 and 100

if data.isdigit(): # check that the string holds a number using a while loop

# if and only if the user has entered a whole number assign guess

guess = int(data)

if guess not in range (1, 101):

print("please enter a valid number.")

counter = 0

continue

if guess > number: # check the guess and report too high or two low

print("Too high, please try again: ")

if guess < number:

print("Too low, please try again: ")

if guess == number:

break

# check number of guesses .. over ten? end the loop

if counter >= 10:

break

# End of Guess Loop

if counter < 10:

print("Correct! you guessed {0} times.".format(counter))

counter = 0

guess = -1

number = random.randint(1,100)

break

elif guess == number:

print("Max guesses reached, but you guessed right at last")

counter = 0

guess = -1

number = random.randint(1,100)

break

else:

print("Sorry it was {0}. You are out of guesses".format(number))

counter = 0

guess = -1

number = random.randint(1,100)

break

data = input("Would you like to play again 'yes' (Y/y) or 'no' (N/n): ") # Ask to play again - process the play flag according to user input

if data.lower() == "y":

play = True

elif data.lower() == "n":

play = False

print("Thanks for playing! bye.")

# End of game loop

print("Game Over .. ")

**Turtle Graphics (Draw Name)**

I added a pensize and changed the color to blue for my own personal customization to what I wanted it to look like and what I thought would be cool. After finishing printing out my name in block letters, I had the problem of centering my name in the 600x600 grid. So I took my highest x which was 100 and my lowest x which was -290 and I added them together and divided the result by two which gave me 195. I then proceeded to subtract that number by my max which gave me the result of 95, so I moved all my points over by a total of 95 spaces but something still seemed off. I forgot to take into consideration the padding on my max value which would make it 120 so I had to restart all those calculations and instead have to move everything by 85 instead, that seems to look a lot better. Now we do the same with the y-axis, so the min I had was 0 and the max I had was 100, so I had to move it down 50 spaces. After this, my max and min x’s were 205, -205 and my max and min y’s were 50, -50 and it was centered but I was missing part of my o so I had to fix that up and it is now working perfectly.

# Mini Programming Project - Turtle Graphics Resource File

# https://docs.python.org/2/library/turtle.html#

#

# Purpose to draw your name to the screen using turtle commands.

# Note: people with long names need only print the first six characters

# Note: How would you perfectly centre your name on the screen?

#

# Author: Cole Becker

#

# Date: September 19, 2023

# setup the environment

# recall the x,y coordinate system is 0,0 in the centre of the drawing window

import turtle

window = turtle.Screen()

window.setup(width=600, height=600) # standardize the size of the screen

pen = turtle.Turtle()

pen.pensize(3)

pen.color('cyan')

padding = 20 # padding between letters

# relocate the turtle near the left side of the screen

pen.penup()

pen.goto(-205,0)

pen.pendown()

# Draw the first letter of your name here C

pen.penup()

pen.goto(-115, 50)

pen.pendown()

pen.goto(-205,50)

pen.goto(-205 ,-50)

pen.goto(-115, -50)

# relocate the pen for the next letter

pen.penup()

x,y = pen.position() # get the x and y position of the turtle

pen.goto(x+padding,0) # set the pen to draw the next letter

pen.pendown()

# Draw the next letter of your name here . . O

pen.goto(-115+padding, -50)

pen.goto(-115+padding, 50)

pen.goto(-15+padding, 50)

pen.goto(-15+padding, -50)

pen.goto(-115+padding, -50)

pen.penup()

pen.goto(-15+padding, -50)

# relocate the pen for the next letter

pen.penup()

x,y = pen.position() # get the x and y position of the turtle

pen.goto(x+padding, 0) # set the pen to draw the next letter

pen.pendown()

# Draw the next letter of your name here . . . L

pen.goto(5+padding, 50)

pen.goto(5+padding, -50)

pen.goto(85+padding, -50)

# relocate the pen for the next letter

pen.penup()

x,y = pen.position() # get the x and y position of the turtle

pen.goto(x+padding,0) # set the pen to draw the next letter

pen.pendown()

# Draw the last letter of your name here . . E

pen.goto(105+padding, 50)

pen.goto(185+padding, 50)

pen.goto(105+padding, 50)

pen.goto(105+padding, 0)

pen.goto(185+padding, 0)

pen.goto(105+padding, 0)

pen.goto(105+padding, -50)

pen.goto(185+padding, -50)

pen.penup()

x,y = pen.position() # get the x and y position of the turtle

pen.goto(600,0) # set the pen to draw the next letter

pen.pendown()

# end of program

window.mainloop() # needed in most programming environments

**Exercise 7.1**

from pcinput import getInteger

num = getInteger("please enter your number: ")

for i in range (1,11):

result = i \* num

print(f"{i} \* {num} is {result}")

**Exercise 7.2**

from pcinput import getInteger

num = getInteger("please enter your number: ")

counter = 1

while counter <= 10:

result = num \* counter

print(f"{counter} \* {num} is {result}")

counter += 1

**Exercise 7.3**

from pcinput import getInteger

total = 10

largest = 0

smallest = 0

divided\_three = 0

for i in range(total):

num = getInteger("enter a numer "+str(i+1)+": " )

if num%3 == 0:

divided\_three += 1

if i == 0:

smallest = num

largest = num

continue

if num < smallest:

smllest = num

if num > largest:

largest = num

print("smallest number is", smallest)

print("largest number is", largest)

print("divisible by three", divided\_three)

**Exercise 7.4**

bottles = 99

s = "s"

while bottles != "no":

print("{0} bottle{1} of beer on the wall, {0} bottle{1} of beer.".format( bottles, s))

bottles -= 1

if bottles == 1:

s = " "

elif bottles == 0:

s = "s"

bottles = "no"

print("Take one down, pass it around, {0} bottle{1} of beer on the wall.".format( bottles, s))

**Exercise 7.5**

num1 = 0

num2 = 1

print( 1 )

while True:

num3 = num2 + num1

if num3 > 1000:

break

print( num3 )

num1 = num2

num2 = num3