CECS 277 – Lab 1 – Python Basics

Shell Game

Create a program that allows the user to play the Shell Game, where a player bets that they can guess the location of the ball under a set of three shells or cups.

The user should start the game with $100. Hide the ball in one of three places by randomizing its location with a value between 1 and 3. Prompt the user to enter an amount to bet (check that the user has enough money, otherwise tell them that it is invalid). Then prompt the user to enter their guess for where the ball is hidden (check that the entered value is between 1 and 3). If it is a match, then the user receives double their bet. Display the location of the ball and tell the user if they were correct or not. Repeat the game until the user runs out of money or decides to quit.

Example Output (user input is in italics)**:**

```
-Shell Game-
Find the ball to double your
bet!

You have $100.
Bet amount? 50

  ___     ___     ___
 /   \   /   \   /   \
/  1  \ /  2  \ /  3  \
------- ------- -------
Make a guess: 1

  ___     ___     ___
 /   \   /   \   /   \
/     \ /  o  \ /     \
------- ------- -------
Sorry... you lose.
Play again? (Y/N): y

You have $50.
Bet amount?  f
Invalid input - should be an
integer.
Bet amount? -1
Invalid input - should be
within range 1-50.
Bet amount? 75
Invalid input - should be
within range 1-50.
Bet amount? 30

  ___     ___     ___
 /   \   /   \   /   \
/  1  \ /  2  \ /  3  \
------- ------- -------
Make a guess: 2

  ___     ___     ___
 /   \   /   \   /   \
/  o  \ /     \ /     \
------- ------- -------
Sorry... you lose.
Play again? (Y/N): y

You have $20.
Bet amount? 10

  ___     ___     ___
 /   \   /   \   /   \
/  1  \ /  2  \ /  3  \
------- ------- -------
Make a guess: 2

  ___     ___     ___
 /   \   /   \   /   \
/     \ /  o  \ /     \
------- ------- -------
Congratulations!

Play again? (Y/N): y

You have $30.
Bet amount? 30

  ___     ___     ___
 /   \   /   \   /   \
/  1  \ /  2  \ /  3  \
------- ------- -------
```

```
Make a guess: 3
  ___    ___    ___
 /   \  /   \  /   \
/  o  \ /     \ /     \
------- ------- -------
```

```
Sorry... you lose.
You're out of money!  Game
over.
```

Notes:

1. Place your name, date, and a brief description in a comment block at the top of your program.
2. Your code should be defined in a main function.
3. Use the check_input module provided on Canvas to check the user's input for invalid values. Add the .py file to your project folder to use the functions. Examples using the module is provided in a reference document on Canvas. Use get_int_range for the bet and the guess, and get_yes_no to see if the user wants to play again.
4. Use the random module to generate your random numbers. Examples for generating random numbers is provided in a reference document on Canvas.
5. No need to create extra functions or add lists to your code, you'll only need the main function with a while loop and some if statements.
6. Please read through the Coding Standards reference document on Canvas for guidelines on how to name your variables and to format your program.
7. Add brief comments in your program to describe sections of code (you should not have a comment describing every single line).
8. Use the escape character '\\' when you want to display a '\'.
9. Thoroughly test your program before submitting:
    a. Make sure the game re-randomizes the location of the ball every round.
    b. Make sure that the user cannot enter an invalid input for the bet (must be between 1 and the user's remaining money).
    c. Make sure that the user cannot enter an invalid input for the guess (1-3).
    d. Make sure that the ball is displayed at the correct location.
    e. Make sure that the game accurately reports whether the user chose the correct location of the ball.
    f. Make sure that the user gains the amount of the bet when they win and takes away the amount of the bet when they lose.
    g. Make sure the game repeats when the user chooses to play again. Game re-randomizes the location of the ball but does not reset their money.
    h. Make sure the game ends when the user is out of money or when the user decides to quit.

Shell Game Rubric – Time estimate: 3 hours

Shell Game **10 points**	Correct. 2 points	A minor mistake. 1.5 points	A few mistakes. 1 point	Several mistakes. 0.5 points	No attempt. 0 points
Guess: 1. Generates a new random number every round in the range 1-3. 2. Uses if statements to check the ball's location when displaying solution. 3. Uses if statements to check if user's guess is the same as the ball's location.					
Bet: 1. Correct amount is added to user's money if user wins. 2. Correct amount is subtracted from user's money if user loses. 3. Uses a while loop that repeats until user quits or runs out of money.					
Input: 1. Checks that the user's guess is a valid integer within the range 1-3. 2. Checks that the user's bet is a valid integer between 1 and user's money. 3. Checks that the user's input is a 'Y' or 'N' when asked to play again. 4. Displays an error message when the user's input is invalid.					
Output: 1. Displays user's money. 2. Prompts for bet and guess. 3. Displays shells with 1, 2, 3. 4. Displays shells with ball at correct location. 5. Correctly displays whether user won or lost. 6. Correctly displays when user is out of money. 7. Prompts user to play again.					
Code Formatting: 1. Code is in a main function. 2. Correct spacing. 3. Meaningful variable names. 4. No global variables. 5. Correctly documented.					