CECS 277 – Lab 1 – Python Basics

Three Card Monte

Create a program that allows the user to play the game Three Card Monte, where a player bets that they can guess the location of the queen in a set of three cards.

The user should start the game with $100. Hide the queen 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 their inputted value is greater than 0 and that the user has enough money, otherwise tell them that it is an invalid bet and allow them to reenter their value. Then prompt the user to enter their guess for where the queen is hidden. Check that their guess is between 1 and 3, otherwise tell them that it is an invalid value and allow them to reenter their guess. If it is a match, then the user wins their bet amount, otherwise they lose. Display the location of the queen and tell the user if won or not. Repeat the game until the user runs out of money or quits the game.

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

```
-Three Card Monte-
Find the queen to double your
bet!

You have $100.
How much you wanna bet? 50
+-----+ +-----+ +-----+
|     | |     | |     |
|  1  | |  2  | |  3  |
|     | |     | |     |
+-----+ +-----+ +-----+
Find the queen: 1
+-----+ +-----+ +-----+
|     | |     | |     |
|  K  | |  Q  | |  K  |
|     | |     | |     |
+-----+ +-----+ +-----+
Sorry... you lose.
Play again? (Y/N): y

You have $50.
How much you wanna bet?  f
Invalid input - should be an
integer.
How much you wanna bet? -1
Invalid input - should be
within range 1-50.
How much you wanna bet? 75
Invalid input - should be
within range 1-50.
How much you wanna bet? 30
+-----+ +-----+ +-----+
|     | |     | |     |
|  1  | |  2  | |  3  |
|     | |     | |     |
+-----+ +-----+ +-----+
Find the queen: 2
+-----+ +-----+ +-----+
|     | |     | |     |
|  Q  | |  K  | |  K  |
|     | |     | |     |
+-----+ +-----+ +-----+
Sorry... you lose.
Play again? (Y/N): y

You have $20.
How much you wanna bet? 10
+-----+ +-----+ +-----+
|     | |     | |     |
|  1  | |  2  | |  3  |
|     | |     | |     |
+-----+ +-----+ +-----+
Find the queen: 2
+-----+ +-----+ +-----+
|     | |     | |     |
|  K  | |  Q  | |  K  |
|     | |     | |     |
+-----+ +-----+ +-----+
You got lucky this time...

Play again? (Y/N): y

You have $30.
How much you wanna bet? 30
+-----+ +-----+ +-----+
|     | |     | |     |
|  1  | |  2  | |  3  |
|     | |     | |     |
+-----+ +-----+ +-----+
Find the queen: 3
+-----+ +-----+ +-----+
|     | |     | |     |
|  K  | |  Q  | |  K  |
|     | |     | |     |
+-----+ +-----+ +-----+
Sorry... you lose.
You're out of money.  Beat it
loser!
```

Notes:

1. Place your name, date, and a brief description in a comment block at the top of your program.
2. 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.
3. Use the random module to generate your random numbers. Examples for generating random numbers is provided in a reference document on Canvas.
4. Your code should be defined in a main function.
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. Thoroughly test your program before submitting:
    a. Make sure that the user's money is not reset to $100 each round.
    b. Make sure the game re-randomizes the location of the queen every round.
    c. Make sure to draw the cards (you don't have to match the example output exactly, but you should draw cards with labels).
    d. Make sure that the user cannot enter an invalid input for the bet (must be between 1 and the user's remaining money).
    e. Make sure that the user cannot enter an invalid input for the guess (1-3).
    f. Make sure that the queen is displayed at the correct location when revealed.
    g. Make sure that the game accurately reports whether the user chose the correct location of the queen.
    h. 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.
    i. Make sure the game ends when the user is out of money or when the user decides to quit.

Three Card Monte Rubric – Time estimate: 2.5 hours

Three Card Monte 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 queen's location when displaying solution. 3. Uses if statements to check if user's guess is the same as queen's location.					
Bet: 1. User begins game with $100. 2. Correct amount is added to user's money if user wins. 3. Correct amount is subtracted from user's money if user loses. 4. 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 cards with 1, 2, 3. 4. Displays cards with queen at correct location. 5. Correctly displays that user won/lost. 6. Correctly displays when user has run 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.					