**Games of Chance**

**Overview**

This project is slightly different than others you have encountered thus far. Instead of a step-by-step tutorial, this project contains a series of open-ended requirements which describe the project you’ll be building. There are many possible ways to correctly fulfill all of these requirements, and you should expect to use the internet , and other resources when you encounter a problem that you cannot easily solve.

**Project Goals**

You will work to write several functions that simulate games of chance. Each one of these functions will use a number of parameters, random number generation, conditionals, and return statements.

**Prerequisites**

**1.**

In order to complete this project, you should have completed the first 3 sections.

This includes the lessons on Syntax, Functions, and Control Flow.

**Project Requirements**

**2.**

The project starts by importing the random module. Every game of chance will involve generating a random number.

For example, to generate a random between 1 and 10 (inclusive) and store it in a variable named num, use this line of code:

num = random.randint(1, 10)

The project also has a variable named money that starts at 100. This represents your current amount of money. In every game of chance, you will be able to bet money. The value of money should change depending on whether you win or lose the game.

At the end of the project, we will have you call several of these functions in a row to see how much money you can win. But it is a good idea to call these functions to test them as you are creating them.

Your functions should have print statements to help the user understand what has happened. For example, in games of chance that involve rolling dice, you should print out the result of those dice rolls. You should also print whether the player won or lost the game, and how much money they won or lost.

**3.**

Create a function that simulates flipping a coin and calling either "Heads" or "Tails". This function (along with all of the other functions you will write in this project) should have a parameter that represents how much the player is betting on the coin flip.

This function should also have a parameter that lets the player call either "Heads" or "Tails".

If the player wins the game, the function should return the amount that they won. If the player loses the game, the function should return the amount that they lost as a negative number.

**4.**

Create a function that simulates playing the game [Cho-Han](https://en.wikipedia.org/wiki/Ch%C5%8D-han). The function should simulate rolling two dice and adding the results together. The player predicts whether the sum of those dice is odd or even and wins if their prediction is correct.

The function should have a parameter that allows for the player to guess whether the sum of the two dice is "Odd" or "Even". The function should also have a parameter that allows the player to bet an amount of money on the game.

**5.**

Create a function that simulates two players picking a card randomly from a deck of cards. The higher number wins.

Once again, this function should have a parameter that allows a player to bet an amount of money on whether they have a higher card. In this game, there can be a tie. What should be returned if there is a tie?

Check the hint to see an additional challenge for this game.

**6.**

Create a function that simulates some of the rules of [roulette](https://en.wikipedia.org/wiki/Roulette). A random number should be generated that determines which space the ball lands on.

When we wrote our function, we allowed the user to guess "Odd", "Even", or a specific number. We also implemented the logic associated with the 0 and 00 spots. For example, the player loses if they guess either "Odd"or "Even" and either 0 or 00 comes up.

Implement as many rules of roulette as you’d like. Make sure to consider the different ways roulette rewards a win. Check the hint to see more about this!

**7.**

Call each of your functions at least once. Below is an example of betting $10 on a coin flip and updating the amount of money you have based on whether you win or lose :

money += coin\_flip("Heads", 10)

Make sure there are enough print statements so you can understand what games were played, what happened during those games, and the amount of money you have after each game is played.

**8.**

Expand your program to check for edge cases. What should happen if a player tries to bet more money than they have? What should happen if a player bets a negative amount of money? What should happen if a player calls "heads" or "Heads!" rather than "Heads".

Try to make it very difficult for someone to break your program.