

Lab 8 – Coin Flipping

Goals

- Familiarity with random module
- Familiarity with if and if/else statements
- Familiarity with loops (while/for)
- Learn how to write and use functions
- Understanding of Python input and output

Setup

- Create a new .py file in your desired directory, and rename the file
- When you name the code, use the following naming convention
ITP115_l#_lastname_firstname
(replace # with this lab number)
- Your new file must begin with comments in the following format (replace the name and email with your actual information):

```
# Name
# ITP 115, Spring 2017
# Lab practical L^ (replace ^ with this Lab number)
# USC email
```

Requirements

Your program must perform the following:

- You will write a program that will run an experiment with tossing a coin.
- Hint 1:
 - o Write the **three functions** in the order they appear here: **coin**, **experiment**, **main**. Each function does a simple operation, but the complexity comes because **main** calls **experiment**, and **experiment** calls **coin**
- Hint 2:
 - o To randomly select an item from a list, use **random.choice(...)**
 - o For example,

```
import random # put this lines at the TOP of your program
myList = [5, 3, 90]
num = random.choice(myList)
#now num is assigned a value randomly selected from [5, 3, 90]
```

- Hint 3:
 - Use a **global constant** called **COIN_FACES** containing the strings **"heads"** and **"tails"**. Recall that since this is a global constant, this variable can be accessed by any of the functions defined in the file.
- Write a function called **coin** that simulates the flip of a coin returning **"heads"** or **"tails"**. Depending on the input, you will make this coin act as a fair (50% heads/50% tails) coin or an unfair coin.
 - *Parameters: weight (integer)*
 - *Returns: either "heads" or "tails" (string)*
 - When you call this function, it will return **"heads"** or **"tails"** in depending on the weight of the coin
 - Give the input parameter (weight) a default value of **50**. This weight represents the percentage of the time **tails** comes up. A fair coin by default has a 50/50 weight for heads/tails.
 - Generate a random number between 1 and 100.
 - Compare the random number to the weight input. Return **"heads"** or **"tails"** accordingly.
- Write a function called **experiment** that simulates the process of flipping coins by calling the **coin** function
 - *Parameters: userWeight*
 - *Returns: int (number of flips it took to get three "heads")*
 - Create a loop that calls the **coin** function repeatedly until you get **"heads"** three times in a row
 - Return the number of flips it took to get three **"heads"** in a row
 - Examples
 - HHH 3 flips
 - THHH 4 flips
 - HHTHHH 6 flips
- Write a function called **main** that runs the **experiment** function
 - *Parameters: none*
 - *Returns: none*
 - Run the experiment 10 times
 - Ask the user if they want to flip a fair coin
 - If they say no, ask them for the weight of the coin in favor of tails (ex: if they want 75% tails, they would enter 75). "Pass" this value into **experiment**
 - Display the average number of flips it takes to get 3 **"heads"** in a row

Sample Output

With a fair coin:

Would you like to use a fair coin (y/n)? **y**

The average for 3 heads in a row is: 17.7

With an unfair coin:

Would you like to use a fair coin (y/n)? **n**

Please enter the desired weight of the coin in favor of tails (ex: if you want 75% heads, enter 75) **9**

The average for 3 heads in a row is: 3.3

Deliverables and Submission Instructions

- A compressed folder (zip file) containing your Python code. This can be done by:
 - a. Windows (*you must find the folder on your computer—this can't be done within PyCharm*):
 - i. Select your lab file
 - ii. Right click
 - iii. Send to ->
 - iv. Compressed (zipped) folder
 - v. Rename this folder with the following name:
ITP115_l#_lastname_firstname
(replace # with this assignment number)
 - vi. Submit this zipped folder through Blackboard
 - b. OSX (*you must find the folder on your computer—this can't be done within PyCharm*):
 - i. Select your lab file
 - ii. Right click
 - iii. Compress 1 item
 - iv. Rename this folder with the following name:
ITP115_l#_lastname_firstname
(replace # with this assignment number)
 - v. Submit this zipped folder through Blackboard