Lab 13 – Objects

Goals

- Practice with objects
- Practice with instance variables
- Practice with instance method

Setup

- Create a new Python project
- Use the following naming convention
  `ITP115_l#_lastname_firstname`
  (replace *lastname* with your last/family name and *firstname* with your first name)
- Your new file must begin with comments in the following format (replace the name and email with your actual information):

```
# Name
# ITP 115, Fall 2017
# Lab L#
# USC email
```

Requirements

Your program must perform the following:

- Write a **`Die`** class and a **`main`** function that creates two **`Die`** objects
- In your **`Die`** class, have the following:
  - Instance attributes / variables (to be assigned inside **`__init__`**)
    - **`rollValue`**: holds value of dice roll
    - **`sides`**: number of sides of dice
  - Constructor method
    - **`__init__(numSides)`**
      - Input arguments (1): **`numSides`**
        - the number of sides you would like your die to have;
        - if no input argument is provided, a **default value** of 6 should be assigned
      - Return value: none
      - Set **`sides`** equal to **`numSides`**

        - Set **`rollValue`** equal to **`0`**
    - Instance methods
        - **`roll`**
            - Input arguments (0): none
            - Return value: the number the die rolled
            - Simulate "rolling" a die by choosing a random number from `1` through **`sides`**
            - Store the number of the die rolled internally to **`rollValue`**
        - **`__str__`**
            - Input arguments (0): none
            - Return value: **`rollValue`** as a string
            - Convert **`rollValue`** to a string and return it
- You will write two additional functions (not part of the class)
    - **`main()`**
        - Ask the user if they want to use the default number of sides
        - If not, then ask the user for the number of sides
        - Create two Die objects
        - Roll each die
        - Use **`__str__`** to print the numbers rolled by each die
        - Pass each roll into **`findSum()`**
        - Print the result of **`findSum()`** to the screen
    - **`calcDieSumAverage(die1, die2, numRolls)`**
        - Input arguments (3): two die objects, and an int that represents how many rolls to perform
        - Output: a float
        - The function will loop **`numRolls`** times and call **`roll()`** on each die. The sum should be stored for all the loops. Return the average.

Sample Output

Output 1:

```
Do you want to use a default dice for your first dice (y/n)? y
Do you want to use a default dice for your second dice (y/n)? y
Dice 1 rolled a 3.  Dice 2 rolled a 5.
How many rolls do you want to average the sum of? 100
The average sum of Dice 1 and Dice 2 is 7.14
```

Output 2:

```
Do you want to use a default dice for your first dice (y/n)? y
Do you want to use a default dice for your second dice (y/n)? n
How many sides would you like for your second dice? 9
Dice 1 rolled a 4.  Dice 2 rolled a 8.
How many rolls do you want to average the sum of? 100
The average sum of Dice 1 and Dice 2 is 9.56
```

Output 3:

```
Do you want to use a default dice for your first dice (y/n)? n
How many sides would you like for your first dice? 8
Do you want to use a default dice for your second dice (y/n)? y
Dice 1 rolled a 7.  Dice 2 rolled a 6.
How many rolls do you want to average the sum of? 2
The average sum of Dice 1 and Dice 2 is 6.87
```

Output 4:

```
Do you want to use a default dice for your first dice (y/n)? n
How many sides would you like for your first dice? 8
Do you want to use a default dice for your second dice (y/n)? n
How many sides would you like for your second dice? 8
Dice 1 rolled a 7.  Dice 2 rolled a 7.
How many rolls do you want to average the sum of? 100
The average sum of Dice 1 and Dice 2 is 8.45
```

Deliverables and Submission Instructions

- A compressed folder (zip file) containing you 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 folder
        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 lab 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 folder
        ii. Right click
        iii. Compress 1 item
        iv. Rename this folder with the following name:
            `ITP115_l#_lastname_firstname`
            (replace # with this lab number)
        v. Submit this zipped folder through Blackboard