Programming on Purpose (CSC 120)
Winter 2024

Lab 4 – Die with Class

1/25/2024

Objectives

- Practice making and using classes and objects.

Preliminary Setup

For this lab, you will be creating an object-oriented (OO) program to play a simple dice game. Start with the following:

1. There will be no starter code.
2. Create a folder for this lab. Then open the folder in VS Code.
3. Add two files to this folder: `main.py`, where you will put the `main()` function, and `die.py`, where you will put the `Die` class.
4. Make sure to add module docstrings to the top of both files.

1 Step 1: Write the Die class

In the `die` module, write the `Die` class. A `Die` object should have two properties/attributes: the number of sides the die has, and the current value showing on the die. These properties, also known as *instance variables*, should be defined and initialized in the constructor.

Add the following methods:

- A constructor (`__init__` method)) that takes a parameter that indicates the number of sides that the particular die should have. Be sure that *each* of the instance variables is given an appropriate value.

  Remember the constructor's job is to initialize the instance variables. So you have two of them to initialize. One of them is easy – you get the initial value from the parameter. But the other doesn't have a parameter for it. But you still need to initialize it. Give it some arbitrary default value that makes sense. Make sure you are not introducing a magic number!

- A *roll* method that rolls the die (causes the current value to be a new random number from 1 to the number of sides). [Note: you can use `randint` or `randrange` from the built-in `random` module; see the Python documentation.] This method should not return anything.

- A *getter* method named `get_value` that returns the value currently showing on the die.

2 Step 2: Write main

In the main module, write the `main()` function to play a simple game according to the following rules:

1. The game uses a D6 and a D12 (a 6-sided and a 12-sided die).
2. On each turn, we roll both dice.
3. The game is won if one of the dice shows a value that's exactly twice the value of the other.
4. Print out both values rolled.
5. The game is over if one of the dice shows a value that is exactly twice the value of the other. Otherwise, roll again.

Your `main` function should create variables for the two dice: each should be an object of type `Die`.

Your program should use the `input` function to ask the user to hit return before each turn. After each turn, the program should check if we've won. If we've won, print a message saying so and end the program.

3 Step 3: Add a default [optional]

Read Section 9.18 in our Python textbook and learn about optional parameters in Python. Then, modify the `Die` class so that the default number of sides is 6. That is, if we ask to create a die object without specifying the number of sides, we will get a D6. Modify `main()` so it uses this default where appropriate.

4 Revise your code

Before turning in any program in this class, remember this mantra:

> Just because it works doesn't mean it's good.

Part of your grade will also come from things like how understandable and readable your code is. You will also be graded on the neatness, presentation, and style of your program code.

Make sure that all modules and functions are documented (even those that you didn't write).

Don't forget to cite who you received help from and include the honor code affirmation in the docstring of each module that you wrote or modified:

```
I affirm that I have carried out my academic endeavors
with full academic honesty. [Your Name]
```

5 How to submit

Submit the project by uploading `Die.py` and `main.py` to Gradescope.

Make sure to add your partner to the submission so that you both have access to it through Gradescope.