**Problem 1: Dice and Cups**

For this problem, you will build classes for dice and cups.

Begin by creating a SixSidedDie class. The class should include the following methods: roll(), getFaceValue(), and \_\_repr\_\_(). For example:

>>> d = SixSidedDie()

>>> d.roll()

3

>>> d.getFaceValue()

3

>>> d

SixSidedDie(3)

Create a TenSidedDie and a TwentySidedDie class. These two class must extend SixSidedDie. They must provide the same functionality. The must not re-implement any code that is not necessary.

Create a Cup class. A cup will hold several dice that may be rolled at once. The cup may hold any number of six-, ten, or twenty- sided dice. For example, we could create a cup with one of each type of die as follows:

>>> cup = Cup(1,1,1)

…or we could create a cup with 3 six-sided dice…

>>> cup = Cup(3,0,0)

By default, the cup will contain one of each type of die.

>>> cup = Cup()

The Cup class should include the following functionality: roll(), getSum(), \_\_repr\_\_(). For example:

>>> cup = Cup(1,2,1)

>>> cup.roll()

28

>>> cup.getSum()

28

>>> cup

Cup(SixSidedDie(3),TenSidedDie(5),TenSidedDie(3),TwentySidedDie(17))

Demonstrate the functionality of your code with test cases.

**Problem 2: Cups and Dice**

For this problem you will use the classes from the previous problem to build a game. The game will work as follows:

1. Greet the user and ask their name.
2. Provide the user with a balance of 100 dollars.
3. Ask them if they would like to play a game.
4. Generate a random number between 1 and 100. This number will be called the goal.
5. Ask the user how much they would like to bet.
6. Ask the user how many of each die they would like to roll.
7. Create a cup filled with dice according to the user’s input.
8. Roll the cup and display the results.
9. If the roll exactly matches the goal the user receives 10x bet added to their balance.
10. Otherwise, if the roll is within 5 of the goal but not over the user receives 5x bet added to their balance.
11. Otherwise, if the roll is within 10 of the goal but not over the user receives 2x bet added to their balance.
12. Report the results to the user. The message should include their name and updated balance.
13. Ask if they would like to play again. If so go to step 3.

Your game should include extensive error checking for cases including, **but not limited to**:

* The user enters negative numbers for a bet.
* The user enters text rather than numbers.
* The user attempts to bet more than the balance.