In this lab, we will learn to:

1. Design a multiclass application
2. Define class methods
3. Understand concept of *self*
4. Implement a simulation of a game

In this lab, we will create a two player game called HighTwo. In this game, each player will roll one 6–sided die and one 10–sided die. The player with the higher sum wins!

To make this game, you will need to implement the following classes.

Die	Player	HighTwoGame
Number of sides Face up value	Name Die x 2	Player x 2
roll getValue __str__ __add__ __gt__	rollDice getDiceValue __str__	playOneGame playManyGames __str__

Start by downloading the DiceGameProgram.py from D2L. Then create a file called DiceGameClasses.py where you will implement the three classes above. Since the HighTwoGame class will implement a Player, and the Player class will implement a Die, it may be easiest to start by creating the Die class.

Dice class:

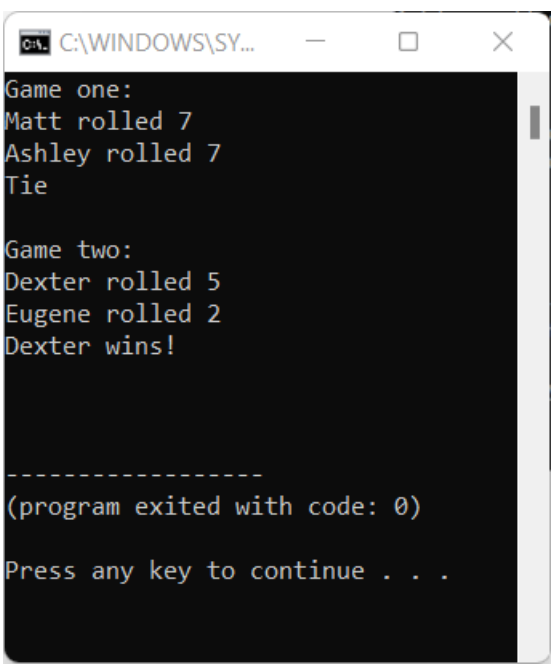
1. You *must* implement the instance variables and methods listed above.
2. You *may* implement any other instance variables and methods that help you build the rest of the game.
3. The constructor should take the number of dice as an argument.
4. Hints:
   (a) Before a die is rolled for the first time, you can assume it has a face up value of 1.
   (b) A die has side numbers between 1 and the number of sides. When you roll a die, one of those numbers *randomly* lands face up.
   (c) __add__ adds the face up value of two dice.
   (d) __gt__ is a comparison operator.

Player class:

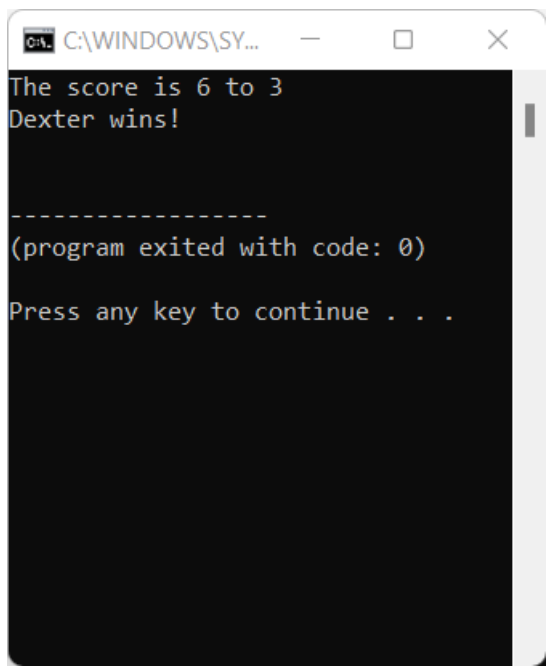
1. You *must* implement the instance variables and methods listed above.
2. You *may* implement any other instance variables and methods that help you build the rest of the game.

HighTwoGame class:

1. You *must* implement the instance variables and methods listed above.
2. You *may* implement any other instance variables and methods that help you build the rest of the game.
3. The constructor should take the names of the players as arguments.
4. Hints:
   - The playOneGame method
     - It should not take any arguments.
     - Get it working correctly before starting the playManyGames.
     - You will need to add one line of code to DiceGameProgram.py.
     - When it's working correctly, you should get output similar to Figure 1: (a) below.
   - The playManyGames method
     - It should take the number of games to play as an argument.
     - This method should internally play the indicated number of games, keep track of how many games each player wins, and report the winner.
     - You will need to write code that calls this method in DiceGameProgram.py.
     - When it's working correctly, you should get output similar to Figure 1: (b) below.



(a) playOneGame



(b) playManyGames

Figure 1: Sample outputs

Once you're done, upload your two .py files to D2L.