**CST-105: Exercise 9 - Dice Game**

The following exercise assesses your ability to do the following:

* Define a model of an object using abstraction and UML
* Translate a model into software by implementing a class
* Utilize class objects in a programming solution

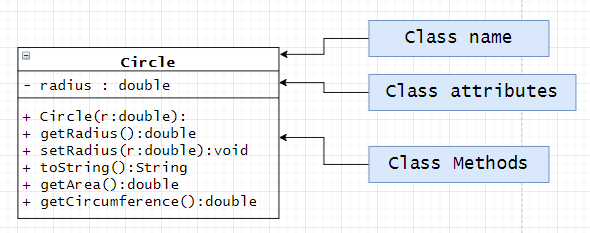
1. Read this document carefully and review the rubric for this assignment before beginning work. Be sure you are familiar with the criteria for successful completion. The rubric link can be found in the digital classroom under the assignment.
2. In this exercise, you will design a class that models a 6-sided die. The die will be used in a game.   
     
   An ADT (abstract data type) is a description of the object you are designing. When designing the ADT, first think of the data that you need to obtain from the die. The only value of interest is the value of the top face. The top face of the die shown here is 6. We will call this the value. Observe that the value of the die is an integer in the range 1–6, inclusive. This can be represented by a single integer.

We begin the ADT with the data that defines the object:  
  
Class data:  
int value

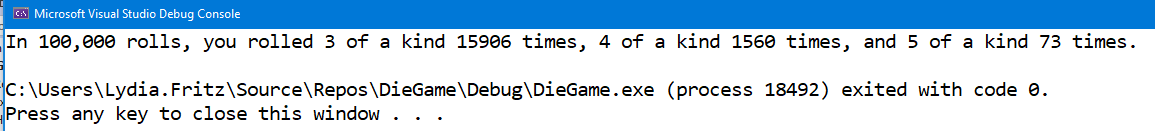
The most important part of the ADT is the functionality. What must a die be able to do? Certainly, we will want to be able to roll the die. When we roll the die, we will want to know the value. We come up with the following function:  
  
//simulate a roll of the die, return a random value from {1,2,3,4,5,6}  
rollDie() : int

Is there anything else a die must be able to do? Allow for more than 6 sides? Be ‘weighted’ so that some outcomes are more likely than others? This is all up to you, the designer.

1. Once you have determined what the Die data type is and how the data type must be able to act in your solution, you can create the UML. The UML has the following format; use the UML shown here as a guide:



1. Once you are satisfied with your UML diagram, begin the implementation of your class. Be sure to include your name and statement of own work at the top of all code files you submit. Be sure to test your Die class as part of the implementation process.
2. Write a driver program that creates an array of 5 Die objects. Roll the 5 die 100,000 times and count the number of times you roll 3 of a kind, 4 of a kind, and 5 of a kind. Here’s a sample output:



1. Create a video in which you present your UML, your class implementation, and your driver program. Videos over 5 minutes in length will not be accepted.
2. Submit the following in the digital classroom:
3. Your Java code for both the driver program and the class implementation
4. Your UML diagram
5. A link to your video