

COP 3330

Programming Assignment 2: Dice Roll Simulation



Program Overview

You will create a Java program that simulates 1000 Dice rolls and creates a histogram of the face value distributions of the dice rolls. If you implement the assignment without using Dice and Die objects, you will receive 0.

Part A

For this part you will create a die (6 faces) class which have those specifications below. The classname will be *Die*

- Die class will have a no argument constructor which will initialize *faceValue* to 1.
- Die class will hold *private faceValue* with integer type as an instance variable.
- You will write a method to get the *faceValue* of the Die.
- You will also write a *void setFaceValue* method to set the *faceValue* of the die using the method arguement which will be an integer.
- Die class will have a void roll method. This method will roll the die randomly, creating a number between 1 to 6, both inclusive. In other words, it will change the *faceValue* randomly. This method will not print anything.

Part B

For this part you will create a Dice class which holds 2 *private Die* objects as instance variables.

- Die class will have a no argument constructor which will initialize *faceValues* of the both dies to 1.
- You will write a method to get the *faceValue* of the Dice which is the sum of the *faceValues* of the 2 dice.
- Dice class will have a *void roll* method. This method will roll both Dice(2 Die objects). This method will not print anything.

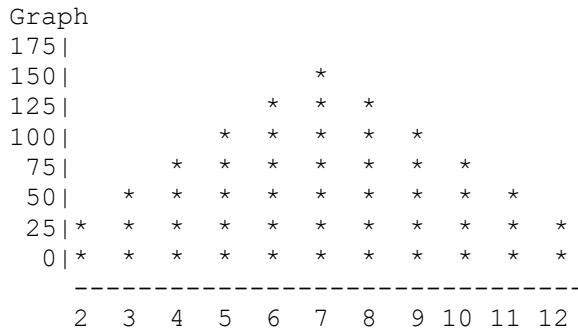
Part C

For this part you will create a DiceTester class which will do the following:
(name of this class will be *DiceTester.java*)

- DiceTester will have the main method inside. The other classes won't have any main method inside.
- You will roll the dice 1000 times. After each roll, you will store number of occurrences of the *faceValues* in an integer array.
- Then you will print those values starting from 2 until 12 after rolling the Dice 1000 times.
- You will also draw a histogram graph shown below. They y axis will start from 175 and decrease by 25 until 0.
- Please take note of the spaces on x axis numbers. If the x axis number has one digit there are 2 spaces between the number and the next number. If the x axis number has 2 digits there is one space between the number and the next number. The histogram should be as pleasing as possible. Do NOT use printf statement while printing anything.

Example output:

```
Number of 2s are 37
Number of 3s are 63
Number of 4s are 89
Number of 5s are 110
Number of 6s are 143
Number of 7s are 160
Number of 8s are 127
Number of 9s are 106
Number of 10s are 77
Number of 11s are 59
Number of 12s are 29
```



Submission Guidelines

- Your Java source file must begin with the following package declaration:

```
package assignment2;
```
- Do not use the default package.
- Include a main method in DiceTester file only. If another file has also a main method, you will lose 20 pts.
- Submit only **three .java** files — do not submit your entire IDE project/class files. Do **not** put the java files inside a folder and zip that folder. There should be no folder structure in your submission. If you fail to do so, you will lose 20 pts.
- Do not use Java Collections or any third-party libraries.
- Your code must compile without errors. Non-compiling submissions will be penalized.
- Your program's output must match with overall format of the expected output. Of course the dice roll numbers will be different in each run.
- Do not concatenate more than one class in your java file. You will receive 0 if you do so.
- All the instance variables stated in this assignment will be private, whereas all the methods will be public.

Evaluation Criteria

Your submission will be assessed based on the following criteria:

10 pts: Proper Indentation (Inconsistent indentation may cost 10 pts)

10 pts: Header comments, package statement and descriptive comments

40 pts: Histogram (No credits if no compilation or no histogram graph)

 Proper spacing between numbers: 20 pts

 Proper spacing between asterisks: 20 pts

10 pts: Instance variables are private, methods are public. Dice roll numbers are written. (All or nothing)

15 pts: Die class is implemented correctly, Method descriptions are followed. Used in Dice class.

15 pts: Dice class is implemented correctly, Method descriptions are followed. Used in DiceTester.