

# ICT 4361 — Java Programming Exercise 2

## Purpose:

This exercise is to provide you an opportunity to use some of the classes we've learned, and to program with arrays and loops.

A number of our exercises this term will be concerned with games. This exercise introduces dice.

A die (plural dice) is a cube, with a different number on each face, with the numbers ranging from one to six.

## What to hand in:

Please hand in a listing for each program requested, formatted in an easy-to-read style.

Ensure your name, and the name of the file is available in a comment at the top of the file.

Also, ensure that you have a sample of the output from the program.

If your program fails to compile, hand in your error listing as your output.

For each question asked, provide one or two sentences summarizing your answer. Please be both complete and succinct.



## Problems:

1. Create a class called `Dice` to represent a single cube. It should have a method called `roll()` that randomly selects a number from 1 to 6 for the value of the dice.
2. Create a test `main` method for the `Dice` class that creates a `Dice`, and rolls it many times. Each time the die is rolled, the `Dice` value should be stored in an array representing how many times that value has come up. After a large number of rolls (say, at least 1,000 -- but 1,000,000 should be no problem), the number of times the die rolled 1, 2, 3, 4, 5, and 6 should be printed.
3. How would you describe the results, and why should that be expected?
4. What would happen if you ran the program many times?
5. Create a class called `DiceStatistics` to represent a pair of dice and the totals of their rolls. It should have an array of two `Dice` as a private data member (these are the `Dice` from Problem 1). It should also have, as a data member, an array of integers to represent the possible totals of rolling two dice. The class should also have the following methods:
  - `initStats()` to initialize all the totals to zero.
  - `rollOnce()` to roll each die once, and add one to the correct totals element
  - `printStatistics()` to print the number of times each total has come up over the number of runs
6. Create a test `main` method for the `DiceStatistics` class that creates a `DiceStatistics`, initializes the statistics, then rolls the dice many times (say, 10000 or so), and then prints the statistics.
7. What is the most likely total, and how likely is it as a percentage?



## Notes

- For the `Dice` class, you will need a random number generator. The way this is done in Java is to use the

- java.util.Random class. After creating an instance, say randomNumber, you can call randomNumber.nextInt(6) to return an integer, randomly distributed from zero to five. You would then add one (since a die goes from one to six, not zero to five) to get the value of the roll.
- Your main test method will have the signature public static void main(String[] args), as is usual, and the method will create a Dice, and invoke roll() on it many times, adding one to the appropriate value each time. Use a variable for the maximum number of times you want to roll() the die, so you can see what happens as you increase the number.
  - DiceStatistics has two arrays in it. One to hold references to the Dice, and the other to hold the counts for the totals from rolling the Dice.

## Evaluation criteria

| Criteria                              | Weight |
|---------------------------------------|--------|
| Dice class                            | 15     |
| Dice main method and output           | 20     |
| Dice question 3                       | 10     |
| Dice question 4                       | 10     |
| DiceStatistics class                  | 15     |
| DiceStatistics main method and output | 15     |
| DiceStatistics question 7             | 10     |