# Laboratory 1

Title of the Laboratory Exercise: Introduction to java simulation

1. **Introduction and Purpose of Experiment**

Computer simulation provides students to design and implement computer simulation models, conduct simulation experiments and evaluate system performance. This laboratory exercise will help the students to get familiar with using object‐oriented simulation in Java.

Java (Structured Parallel Discrete Event Simulation in Java) system is designed to incorporate the parallel programming technology into discrete event simulations. The java system adopts the approach of augmenting a general-purpose language with essential constructs to support simulation modeling based on the process-oriented modeling technology

1. **Aim and Objectives**

Aim

* To use Netbeans and understand using object‐oriented simulation in Java

**Objectives**

At the end of this lab, the student will be able to

* Explain the features and use of Netbeans IDE to develop java programs for simulation
* Edit, compile and execute java programs successfully using Netbeans IDE

1. **Experimental Procedure**

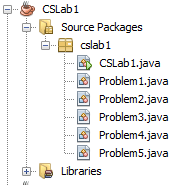
Students are given a set of programs for generating random numbers using built-in methods. Programs should be edited, compiled and executed using Netbeans IDE.

Random number generation using inbuilt methods/manually

Ex: coin toss, die, and cards

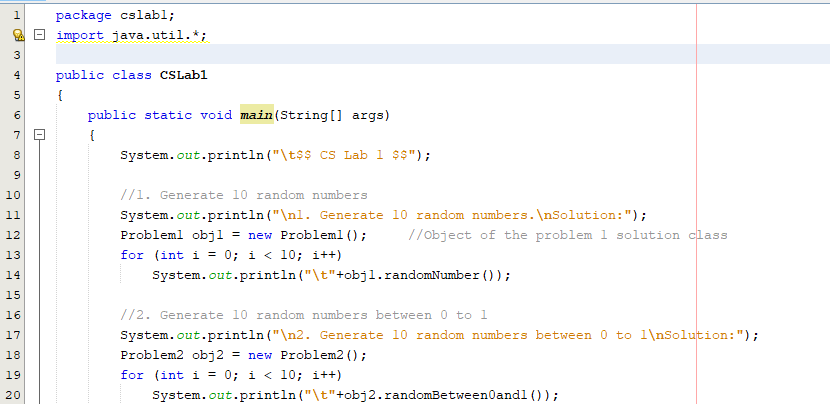
1. **Calculations/Computations/Algorithms**
   1. Develop and implement a java program to generate 10 random number
   2. Develop and implement a java program to generate 10 random numbers in between 0 to 1
   3. Develop and implement a java program to generate 10 normally distributed random numbers in between 0 to 1.
   4. Develop and implement a java program to flip a coin 50 time, and count number of occurrence of head and tail and determine probability distribution of head and tail.
   5. Develop and implement a java program to through a dice 200 times, and count number of occurrence of each face (1,2 …. 6) and determine probability distribution.
2. **Presentation of Results**

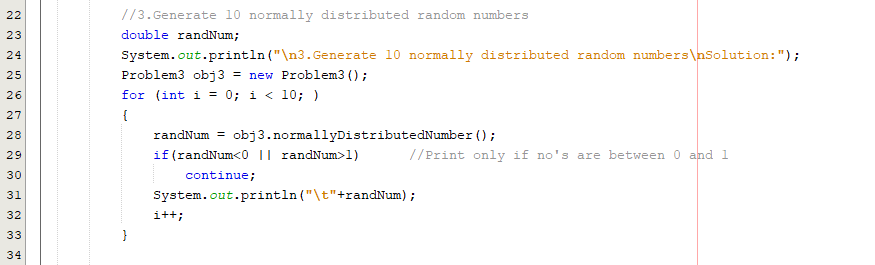
The following is the project structure:

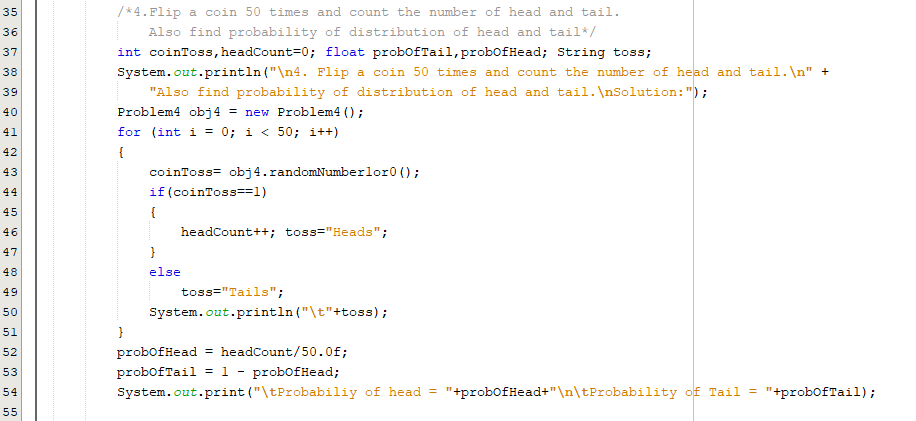


Problem **1** to **5** are the java files to solve the problems from **a** to **e** respectively. The **CSLab1.java** creates the objects of these classes and calls them accordingly.

*CSLab1.java :*







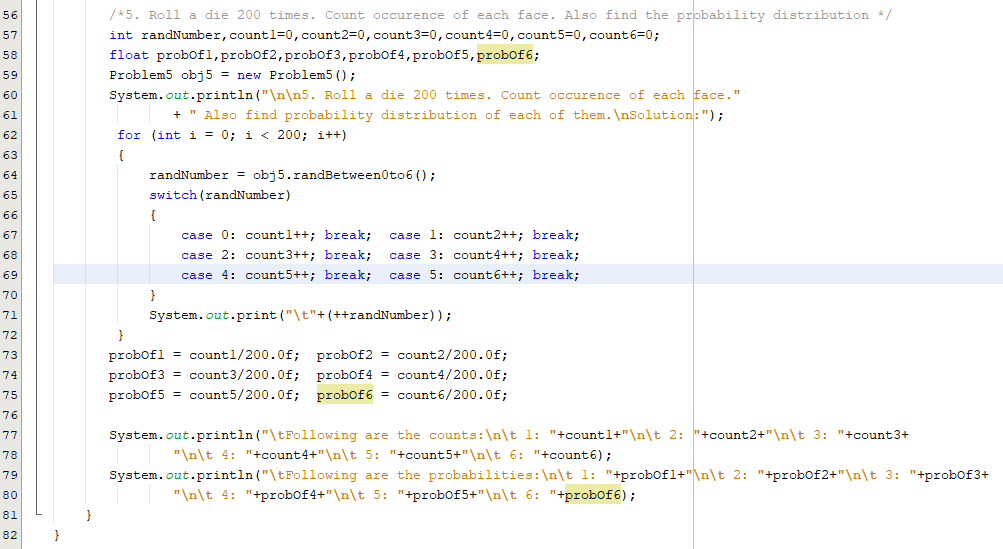


Figure 1 Code containing main( ) function

*Problem 1.java :*

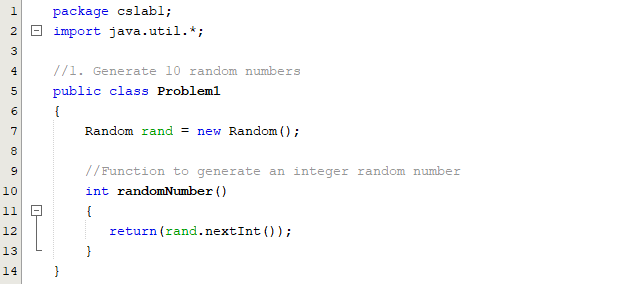


Figure 2 Code to solve problem “a”

*Problem 2.java :*

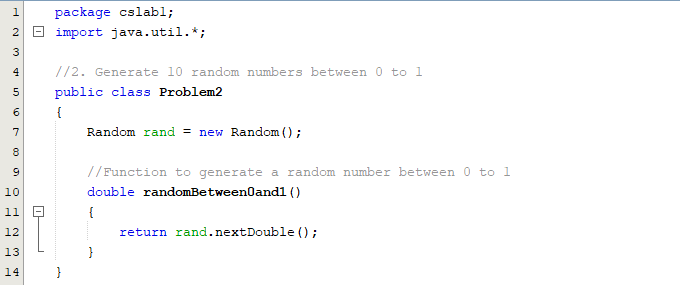


Figure 3 Code to solve problem “b”

*Problem 3.java :*

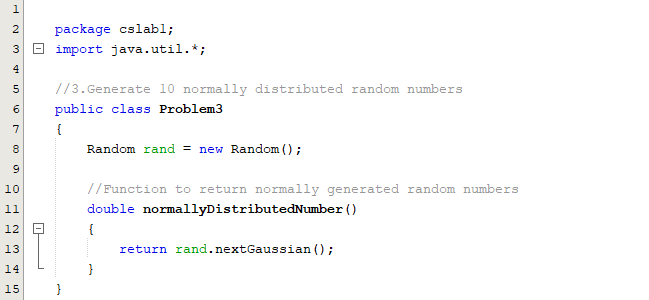


Figure 4 Code to solve problem “c”

*Problem 4.java :*

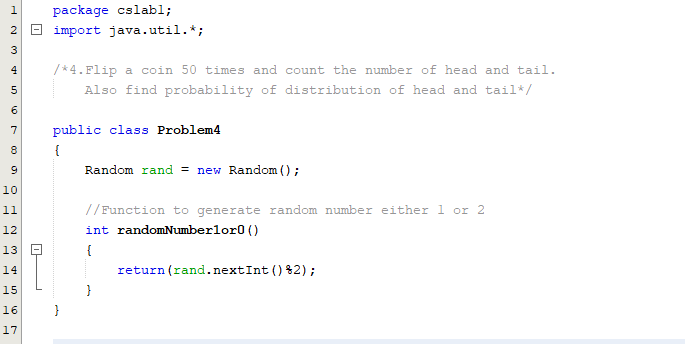


Figure 5 Code to solve problem “d”

*Problem 5.java :*

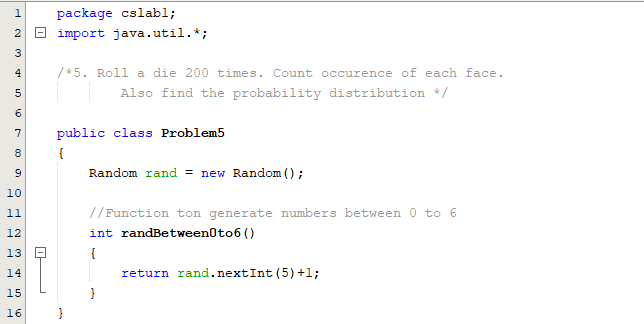
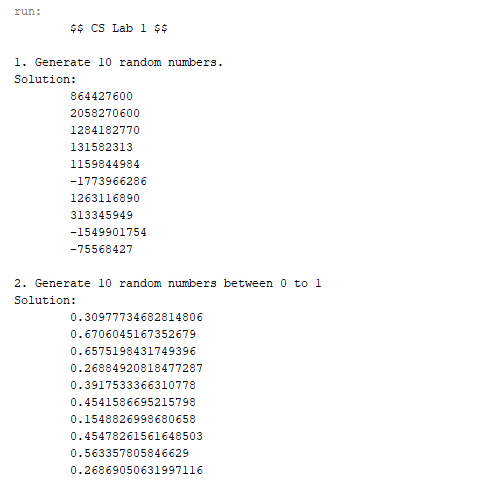
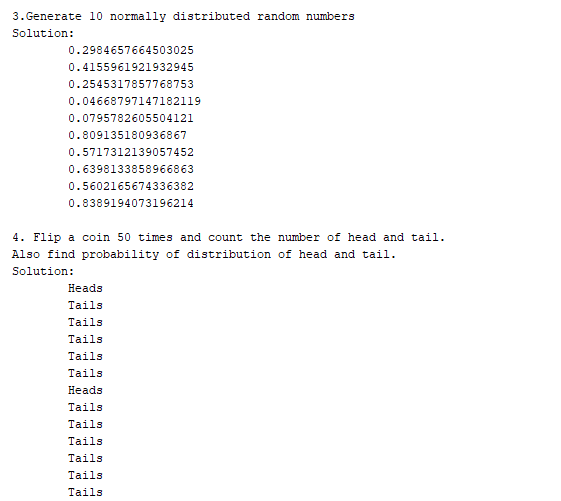


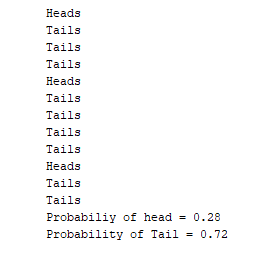
Figure 6 Code to solve problem “e”

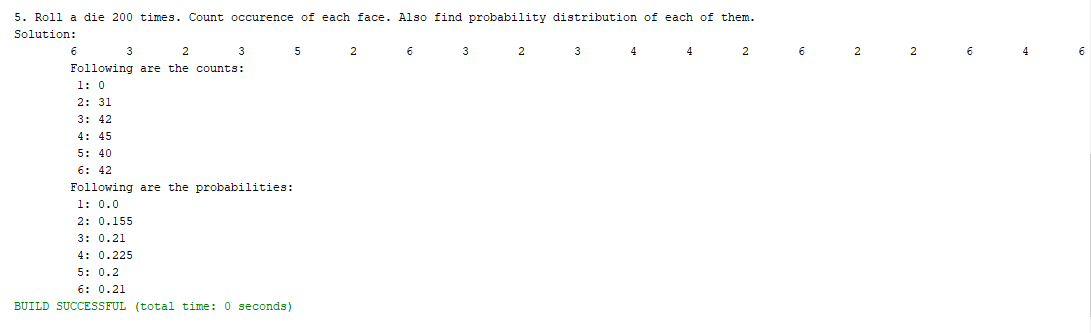
*Output:*

The objects of all the classes are created in **CSLab1.java** file and executed in the main( ) function as seen in the above figures.









1. **Analysis and Discussions**

Random number generation is one of the main phenomenon in computer simulation. In real world, there are many ways to generate a random number but when it comes to computer, the system has to use a predefined formula to do so.

Since the computer uses a mathematical formula to generate the random number, it needs a **seed** using which a series of random numbers are generated. If the value of seed is compromised, then the upcoming random numbers can easily be guessed.

In the current program, the java random number generation function **rand( )** from **java.util.Random** is used. In problem “e”, normally distributed random numbers are generated using **nextGaussian( )** function from the **Random** class.

1. **Conclusions**

Random numbers are very essential in computer simulation as we can generate as many varied numbers to test our mathematical model thus simulating the real world scenario.

1. **Comments**

**1. Limitations of Experiments**

The Random interface functions do not give much control as to on what conditions the random numbers have to be generated.

**2. Limitations of Results**

All the objectives of the current laboratory can be seen in the program developed.

**3. Learning happened**

After this laboratory, one can learn the importance of random numbers generation in simulating a real world scenario. Also, proficient knowledge can be obtained on choosing only certain set of random numbers based on the condition required by the currently simulating scenario.

**4. Recommendations**

None.