S.No: 14	Exp. Name: ***Multi Threaded Application with Three Threads***	**Date:2025-10-09**

Aim:

Write a Java program that demonstrates multithreading where:

1. A **RandomNumberThread** generates 3 random integers (seeded with 50)
2. For each generated number:
   If the number is even, create a **SquareThread** to calculate its square.
   If the number is odd, create a **CubeThread** to calculate its cube.
3. Each operation should be printed with a 1-second delay between numbers.

Input Format:

- No direct input is required. The random number generator is seeded with 50 (fixed sequence).

Output Format:

```
Random Integer generated : <number1>
Square/Cube of <number1> = <result1>
[1-second delay]
Random Integer generated : <number2>
Square/Cube of <number2> = <result2>
[1-second delay]
Random Integer generated : <number3>
Square/Cube of <number3> = <result3>
```

Example Output:

```
Random Integer generated : 42
Square of 42 = 1764
Random Integer generated : 29
Cube of 29 = 24389
Random Integer generated : 74
Square of 74 = 5476
```

Constraints:

- The RandomNumberThread must generate exactly 3 numbers.
- Use Thread.sleep(1000) for the 1-second delay.
- Handle thread creation and synchronization implicitly (no explicit synchronization needed).

Note:

- Set the seed value to 50 for consistency.
- The print statements for the output are already present in the editor. Your task is to implement the logic in the specified sections of the code.
- Refer to the visible test cases to strictly match the input/output layout.

Source Code:

ClassMthread.java

```
import java.util.Random;
class RandomNumberThread extends Thread {
   public void run() {
      Random random = new Random();
      random.setSeed(50);
      for(int i = 0; i<3; i++){
         int num = random.nextInt(100);

            synchronized(System.out){
               System.out.println("Random Integer generated : " + num);
            }
         Thread t;
         if(num%2==0)
            t = new SquareThread(num);
         else
            t = new CubeThread(num);

         t.start();

         try{
            t.join();
            Thread.sleep(1000);
         }catch(InterruptedException e){
            System.out.println(e);
         }
      }


   }
}

class SquareThread extends Thread {

   int num;
   SquareThread(int num){
      this.num = num;
   }
   public void run(){
      System.out.println("Square of " + num + " = " + (num*num));
   }

}

class CubeThread extends Thread {

   int num;
   CubeThread(int num){
      this.num = num;
   }
   public void run(){
      System.out.println("Cube of " + num + " = " + (num*num*num));
   }
}

public class ClassMthread {
   public static void main(String[] args){
      RandomNumberThread t = new RandomNumberThread();
      t.start();
   }
```

```
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
Random Integer generated : 17
Cube of 17 = 4913
Random Integer generated : 88
Square of 88 = 7744
Random Integer generated : 93
Cube of 93 = 804357
```