//MULTITHREADING

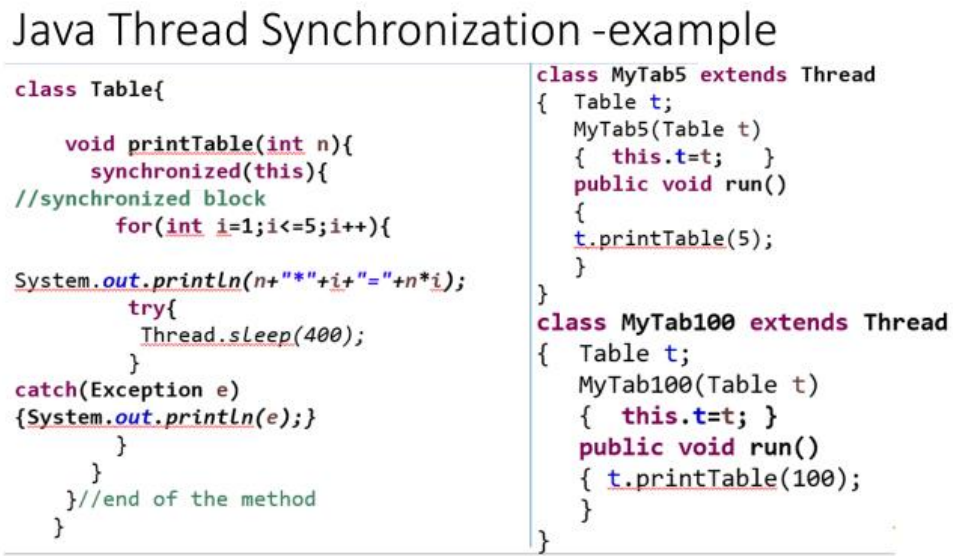
Write a Java program that implements a multi-threaded program which has three threa ds. First thread generates a random integer every 1 second. If the value is even, second thread computes the square of the number and prints. If the value is odd the third thre ad will print the value of cube of the number.

```
import java.util.*;
// class for Even Number
class EvenNum implements Runnable
       {
    public int a;
    public EvenNum(int a)
        {
        this.a = a;
         }
    public void run()
        {
        System.out.println("The Thread "+ a +" is EVEN and Square of
" + a + " is : " + a * a);
    } }

// class for Odd Number
class OddNum implements Runnable {
    public int a;
    public OddNum(int a)  {
        this.a = a;
    }
    public void run()  {
        System.out.println("The Thread "+ a +" is ODD and Cube of "
+ a + " is: " + a * a * a);
    }
}
// class to generate random number
class RandomNumGenerator extends Thread
 {
    public void run()
        {
        int n = 0;
        Random rand = new Random();
        try  {
            for (int i = 0; i < 10; i++)
              {
                n = rand.nextInt(20);
                System.out.println("Generated Number is " + n);
```

```
                // check if random number is even or odd
                if (n % 2 == 0) {
                    Thread thread1 = new Thread(new EvenNum(n));
                    thread1.start();
                }
                else {
                    Thread thread2 = new Thread(new OddNum(n));
                    thread2.start();
                }
        // thread wait for 1 second
        Thread.sleep(1000);
        System.out.println("-----------------------------------");
            }
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
// Driver class/Main class
public class MultiThreadRandOddEven
{
    public static void main(String[] args)
 {
        RandomNumGenerator rand_num = new RandomNumGenerator();
        rand_num.start();
}
}
```

```
C:\StudyGlance\Java_Lab_Programs\W5>javac MultiThreadRandOddEven.java

C:\StudyGlance\Java_Lab_Programs\W5>java MultiThreadRandOddEven
Generated Number is 1
The Thread 1 is ODD and Cube of 1 is: 1
-----------------------------------
Generated Number is 4
The Thread 4 is EVEN and Square of 4 is : 16
-----------------------------------
Generated Number is 1
The Thread 1 is ODD and Cube of 1 is: 1
-----------------------------------
Generated Number is 4
The Thread 4 is EVEN and Square of 4 is : 16
-----------------------------------
Generated Number is 16
The Thread 16 is EVEN and Square of 16 is : 256
-----------------------------------
Generated Number is 8
The Thread 8 is EVEN and Square of 8 is : 64
-----------------------------------
Generated Number is 0
The Thread 0 is EVEN and Square of 0 is : 0
-----------------------------------
Generated Number is 12
The Thread 12 is EVEN and Square of 12 is : 144
```

Q2 : Write a java program to implement Thread Synchronization



Java Thread Synchronization -example

```
class Table{

    void printTable(int n){
      synchronized(this){
//synchronized block
        for(int i=1;i<=5;i++){

System.out.println(n+"*"+i+"="+n*i);
         try{
          Thread.sleep(400);
         }
catch(Exception e)
{System.out.println(e);}
        }
      }
    }//end of the method
   }
```

```
class MyTab5 extends Thread
{  Table t;
   MyTab5(Table t)
   {  this.t=t;   }
   public void run()
   {
   t.printTable(5);
   }
}
class MyTab100 extends Thread
{  Table t;
   MyTab100(Table t)
   {  this.t=t; }
   public void run()
   { t.printTable(100);
   }
}
```

Java Thread Synchronization -example

```
public class TestSync{
   public static void main(String args[]){
   Table obj = new Table();
     //only one object
   MyTab5 t1=new MyTab5(obj);
   MyTab100 t2=new MyTab100(obj);
   t1.start();
   t2.start();
   }
   }
```

```
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
100*1=100
100*2=200
100*3=300
100*4=400
100*5=500
```