**Experiment – 14**

**Multi-Threading**

**Date Of Submission:** 6-11-2020

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

**Concepts Used:** Multithreading

**Algorithm:**

Class Square implements Runnable

Data Members

i : int

t : Thread

Methods:

Override run method in Runnable interface

Start

int sq = i\*i

Print “Square = i”

Stop

Square (a) //constructor

Start

i = a

t = new Thread(this,”Square”)

Stop

Class Cube implements Runnable

Data Members

i : int

t : Thread

Methods:

Override run method in Runnable interface

Start

int cu = i\*i\*i

Print “cube = cu”

Stop

Cube (a) //constructor

Start

i = a

t = new Thread(this,”Square”)

Stop

Class Random implements runnable

Data Members

i: int

t: Thread

Methods:

Random: //constructor

Start

t = new Thread(this,”Random number generator”)

Stop

Override run from Runnable:

Start

i = Math.random()\*100

if(i%2==0) then

new Square(i).t.start()

else

new Cube(i).t.start()

endif

Stop

**Program Code:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* A Multi-threaded Program which has 3 threads,

\* Thread 1: prints a random number every 1 Second;

\* Thread 2: If the number is even then print it's square

\* Thread 3: If the number is odd then print the cube

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Done By: Rohit Karunakaran

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class SquareThread implements Runnable

{

int i;

public Thread t;

public SquareThread(int a)

{

t = new Thread(this,"Square Thread");

i = a;

}

public void run()

{

int sq = i\*i;

System.out.println("The square of the number "+i+" is "+sq);

}

}

class CubeThread implements Runnable

{

public Thread t;

int i;

public CubeThread(int a)

{

t = new Thread(this,"Cube Thread");

i = a;

}

public void run()

{

int qube = i\*i\*i;

System.out.println("The Cube of the number "+i+" is "+qube);

}

}

class RandomThread implements Runnable

{

int i;

public Thread t;

public RandomThread()

{

i = (int)Math.random()\*100;

t = new Thread(this,"Random Number");

}

public void run()

{

for(int j=0;j<10;j++)

{

//Generates a random number from 0-99

i = (int)(Math.random()\*100);

System.out.println("Random Number : "+i);

if(i%2==0)

{

//Square thread

new SquareThread(i).t.start();

//s.t.start();

}

else

{

//cubeThread

new CubeThread(i).t.start();

//c.t.start();

}

try

{

Thread.sleep(1000);

}

catch(InterruptedException e)

{

System.out.println("Interrupted");

}

}

}

}

public class RandomNumber

{

public static void main(String args[])

{

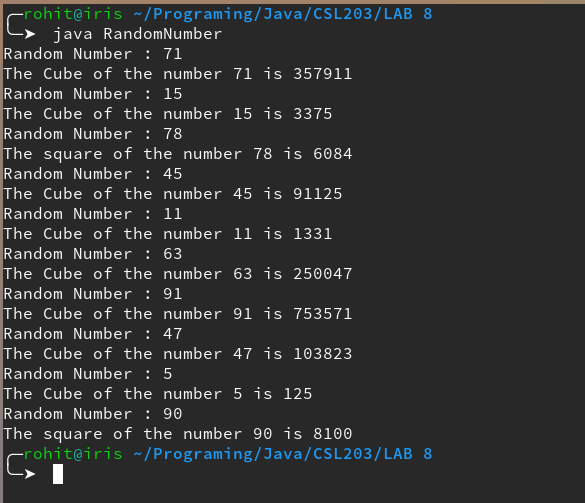
RandomThread r =new RandomThread();

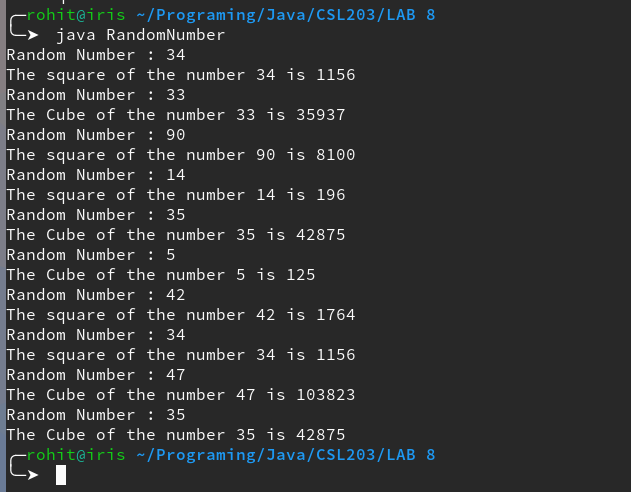
r.t.start();

}

}

**Sample Input/Output**





**Experiment – 15**

**Thread Synchronization**

**Date Of Submission:** 6-11-2020

**Aim:** Write a Java program that shows thread synchronization.

**Concepts used:** Thread Synchronization

**Algorithm:**

Class Test

Methods:

test (String msg)

Print “[”

Thread.sleep(1000)

Print “]”

Stop

Class ThreadInSync implements Runnable

Data Members

msg: String

target : Test

t : Thread

Methods

ThreadInSync (a : Test, s: String)

Start

target = a

msg = s

t = new Thread(this)

Stop

Override run from Runnable interface

Start

target.test(msg)

Thread.sleep(2000)

synchronized (this) do

target.test(msg)

end synchronised

Stop

Class Main

Static Method

main()

Start

test t = new test

ThreadInSync t1 = new ThreadInSync(t,”Thread”)

ThreadInSync t2 = new ThreadInSync(t,”synchronized”)

ThreadInSync t1 = new ThreadInSync(t,”Hello”)

t1.start()

t2.start()

t3.start()

Stop

**Program Code:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Java Program to Demonstrate Thread Synchronization

\* Done By: Rohit Karunakaran

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Test

{

void test(String msg)

{

System.out.print("["+msg);

try{

Thread.sleep(1000);

}

catch(InterruptedException e)

{

System.out.println("Interrupted");

}

System.out.println("]");

}

}

class ThreadsInSync implements Runnable

{

String msg;

Test target;

Thread t;

public ThreadsInSync(Test targ,String s)

{

target = targ;

msg = s;

t= new Thread(this);

}

public void run()

{

target.test(msg);

try{

Thread.sleep(2000);

}

catch(InterruptedException e)

{

System.out.println("Interrupted");

}

synchronized(target){

target.test(msg);

}

}

}

public class Synch

{

public static void main(String args[])

{

Test target = new Test();

ThreadsInSync ob1 = new ThreadsInSync(target,"Hello");

ThreadsInSync ob2 = new ThreadsInSync(target,"Synchronized");

ThreadsInSync ob3 = new ThreadsInSync(target,"Thread");

System.out.println("Without Sychronization");

ob1.t.start();

ob2.t.start();

ob3.t.start();

try{

Thread.sleep(2900);

System.out.println("With Sychronization");

}

catch(InterruptedException e)

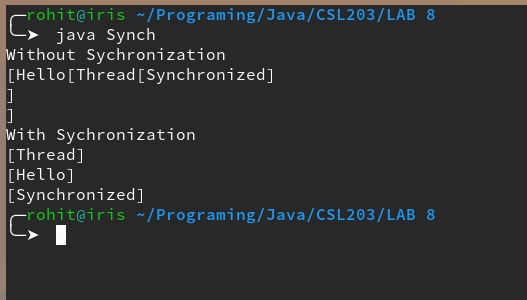
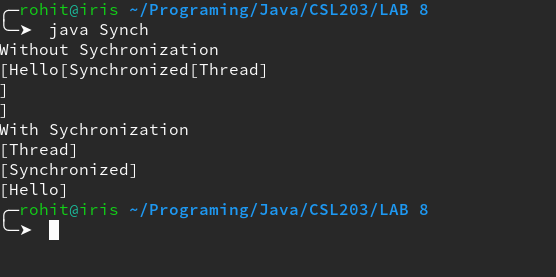
{

System.out.println("Interrupted");

}

}

}

**Sample Input/Output:**