**Date:3/11/2023 Experiment no 8**

**Multi Threading**

**Aim:**

To develop a java program for the given problem statement by using Multi Threading.

**Algorithm:**

Step 1: Start

Step 2 : Initialize Threads: Threads for number generation and calculations are set up.

Step 3 : Run Number Generation: The program starts generating random numbers in an infinite loop.

Step 4 : Generate Number: Random number between 0 and 99 is generated.

Step 5: Check Parity: Check if the number is even or odd.

Step 6: Calculate Square or Cube:

- If the number is even, compute its square.

- If the number is odd, compute its cube.

Step 7: Display Result: Print the calculated square or cube of the number.

Step 8 : Delay and Repeat: Pause for one second, then loop back to generating the next number.

Step 9: Infinite Loop: Continuously generate numbers, perform calculations, and display results until manually stopped.

Step 10: Stop .

**Program:**

import java.util.Random;

class NumberGenerator implements Runnable {

public void run()

{

Random random = new Random();

while (true) {

int number = random.nextInt(100);

System.out.println("Generated Number: " + number);

if (number % 2 == 0) {

Thread squareThread = new Thread(new SquareCalculator(number));

squareThread.start();

} else {

Thread cubeThread = new Thread(new CubeCalculator(number));

cubeThread.start();

}

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

class SquareCalculator implements Runnable {

int number;

SquareCalculator(int number) {

this.number = number;

}

public void run() {

int square = number \* number;

System.out.println("Square of " + number + " is: " + square);

}

}

class CubeCalculator implements Runnable {

int number;

CubeCalculator(int number) {

this.number = number;

}

public void run() {

int cube = number \* number \* number;

System.out.println("Cube of " + number + " is: " + cube);

}

}

public class Multithreading {

public static void main(String[] args) {

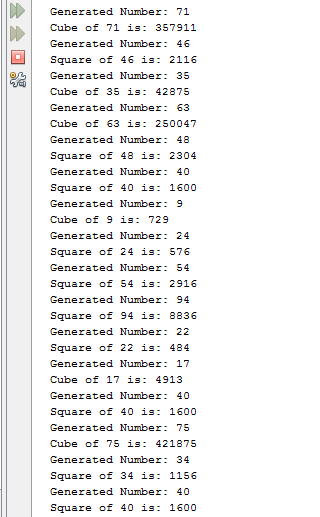
Thread numberGeneratorThread = new Thread(new NumberGenerator());

numberGeneratorThread.start();

}

}

**Output:**



**Result:**

The given Java program was successfully executed and the output was verified.

|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |