VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Programme	:	**BTech. CSE Core**	Semester	:	**Win 2021-22**
Course	:	**Java Programming**	Code	:	**CSE1007**
Faculty	:	**Dr. Pradeep K**	Slot	:	**L9+L10**
Name	:	**Hariket Sukesh Kumar Sheth**	Register No.	:	**20BCE1975**

1. Deduce a Java Program to create a Thread using Thread Class

```java
package lab6;
import java.util.*;
public class Lab6 extends Thread{
    private String s;
    Lab6(String s){
        this.s = s;
    }
    public void run(){
        System.out.println("Thread is Running.");
        if(s.equals("Thread1"))
            for(int i=1; i<=10; i++)
                System.out.println("Thread Running:- "+s+" - Thread "+i);
        if(s.equals("Thread2"))
            for(int i=10; i>=1; i--)
                System.out.println("Thread Running:- "+s+" - Thread "+i);
    }
    public static void main(String args[]){
        Lab6 l1=new Lab6("Thread1");
        Lab6 l2=new Lab6("Thread2");
        l1.start();
        l2.start();
     }
}
```

OUTPUT:

```
Output - Lab6 (run)
run:
Thread is Running.
Thread is Running.
Thread Running:- Thread2 - Thread 10
Thread Running:- Thread1 - Thread 1
Thread Running:- Thread1 - Thread 2
Thread Running:- Thread2 - Thread 9
Thread Running:- Thread1 - Thread 3
Thread Running:- Thread2 - Thread 8
Thread Running:- Thread1 - Thread 4
Thread Running:- Thread1 - Thread 5
Thread Running:- Thread2 - Thread 7
Thread Running:- Thread1 - Thread 6
Thread Running:- Thread1 - Thread 7
Thread Running:- Thread2 - Thread 6
Thread Running:- Thread1 - Thread 8
Thread Running:- Thread2 - Thread 5
Thread Running:- Thread1 - Thread 9
Thread Running:- Thread1 - Thread 10
Thread Running:- Thread2 - Thread 4
Thread Running:- Thread2 - Thread 3
Thread Running:- Thread2 - Thread 2
Thread Running:- Thread2 - Thread 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Deduce a Java Program to create a Thread using Runnable Interface

```java
package lab6;
import java.util.*;
public class Lab6 implements Runnable{
    private String s;
    Lab6(String s){
        this.s = s;
    }
    public void run(){
        System.out.println("Thread is Running.");
        if(s.equals("Thread1"))
            for(int i=1; i<=10; i++)
                System.out.println("Thread Running:- "+s+" - Thread "+i);
        if(s.equals("Thread2"))
            for(int i=10; i>=1; i--)
                System.out.println("Thread Running:- "+s+" - Thread "+i);
    }
    public static void main(String args[]){
        Lab6 l1=new Lab6("Thread1");
        Lab6 l2=new Lab6("Thread2");
        Thread t1 = new Thread(l1);
        Thread t2 = new Thread(l2);
        t1.start();
        t2.start();
     }
}
```

OUTPUT:

```
Output - Lab6 (run)
run:
Thread is Running.
Thread is Running.
Thread Running:- Thread2 - Thread 10
Thread Running:- Thread1 - Thread 1
Thread Running:- Thread1 - Thread 2
Thread Running:- Thread2 - Thread 9
Thread Running:- Thread2 - Thread 8
Thread Running:- Thread1 - Thread 3
Thread Running:- Thread1 - Thread 4
Thread Running:- Thread2 - Thread 7
Thread Running:- Thread1 - Thread 5
Thread Running:- Thread2 - Thread 6
Thread Running:- Thread1 - Thread 6
Thread Running:- Thread2 - Thread 5
Thread Running:- Thread1 - Thread 7
Thread Running:- Thread2 - Thread 4
Thread Running:- Thread2 - Thread 3
Thread Running:- Thread1 - Thread 8
Thread Running:- Thread2 - Thread 2
Thread Running:- Thread2 - Thread 1
Thread Running:- Thread1 - Thread 9
Thread Running:- Thread1 - Thread 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Write a Java program create a list of numbers and then sort in ascending order as well as in descending order simultaneously using threads.

```java
package lab6;

class Ascending extends Thread{
        public void run(){
        System.out.println("Thread is Running.");
            for(int i=1; i<=10; i++)
                System.out.println("Thread Running:- "+"Thread "+i);
    }
}

class Descending extends Thread{
        public void run(){
        System.out.println("Thread is Running.");
            for(int i=10; i>=1; i--)
                System.out.println("Thread Running:- "+"Thread "+i);
    }
}
public class Lab6 extends Thread{
    public static void main(String args[]){
        Ascending a1=new Ascending();
        Descending d1=new Descending();
        a1.start();
        d1.start();
     }
}
```

OUTPUT:

```
Output - Lab6 (run)
run:
Thread is Running.
Thread is Running.
Thread Running:- Thread 1
Thread Running:- Thread 2
Thread Running:- Thread 10
Thread Running:- Thread 9
Thread Running:- Thread 3
Thread Running:- Thread 8
Thread Running:- Thread 7
Thread Running:- Thread 6
Thread Running:- Thread 5
Thread Running:- Thread 4
Thread Running:- Thread 3
Thread Running:- Thread 2
Thread Running:- Thread 1
Thread Running:- Thread 4
Thread Running:- Thread 5
Thread Running:- Thread 6
Thread Running:- Thread 7
Thread Running:- Thread 8
Thread Running:- Thread 9
Thread Running:- Thread 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Write a Java program which first generates a set of random numbers and positive even, positive odd numbers display concurrently.

```java
package lab6;
import java.util.*;

class R1 implements Runnable {
    public void run() {
        int[] randomNumbers = new int[25];
        for (int i = 0; i < randomNumbers.length; i++) {
            randomNumbers[i] = (int)(Math.random() * 99);
            int a = randomNumbers[i];

            if (a % 2 == 0) {
                System.out.println("Even Numbers: " + i);
            }
        }
    }

}
class R2 implements Runnable {
    public void run() {
        int[] randomNumbers = new int[25];
        for (int i = 0; i < randomNumbers.length; i++) {
            randomNumbers[i] = (int)(Math.random() * 99);
            int a = randomNumbers[i];

            if (a % 2 != 0) {
                System.out.println("Odd Numbers: " + i);
            }
        }

    }
}
class R3 implements Runnable {
    public void run() {
        int[] randomNumbers = new int[25];
        for (int i = 0; i < randomNumbers.length; i++) {
            randomNumbers[i] = (int)(Math.random() * 99);
            int a = randomNumbers[i];

            if (a < 0) {
                System.out.println("Negative Number: " + i);
            }
        }
    }
}
class R4 implements Runnable {
    public void run() {
        int[] randomNumbers = new int[25];
        for (int i = 0; i < randomNumbers.length; i++) {
            randomNumbers[i] = (int)(Math.random() * 99);
            int a = randomNumbers[i];
            if (a > 0) {
                System.out.println("Positive Number: " + i);
            }
        }
    }
}
public class Lab6 {
    public static void main(String[] args) {
        Runnable r1 = new R1();
        Thread t1 = new Thread(r1);
        Runnable r2 = new R2();
        Thread t2 = new Thread(r2);
```

```
        Runnable r3 = new R3();
        Thread t3 = new Thread(r3);
        Runnable r4 = new R4();
        Thread t4 = new Thread(r4);
        t1.start();
        t2.start();
        t3.start();
        t4.start();

    }
}
```

OUTPUT:

```
Output - Lab6 (run)
run:
Positive Number: 0
Even Numbers: 2
Odd Numbers: 1
Even Numbers: 3
Positive Number: 1
Even Numbers: 4
Odd Numbers: 4
Even Numbers: 5
Positive Number: 2
Even Numbers: 6
Odd Numbers: 5
Even Numbers: 10
Positive Number: 3
Even Numbers: 14
Odd Numbers: 6
Even Numbers: 17
Positive Number: 4
Even Numbers: 19
Odd Numbers: 7
Even Numbers: 21
Positive Number: 5
Odd Numbers: 8
Positive Number: 6
Odd Numbers: 9
Positive Number: 7
Odd Numbers: 10
Positive Number: 8
Odd Numbers: 12
Odd Numbers: 15
Positive Number: 9
Odd Numbers: 16
Positive Number: 10
Odd Numbers: 17
Odd Numbers: 22
Positive Number: 11
Positive Number: 12
Odd Numbers: 24
Positive Number: 13
Positive Number: 14
Positive Number: 15
Positive Number: 16
Positive Number: 17
Positive Number: 18
Positive Number: 19
Positive Number: 20
Positive Number: 21
Positive Number: 22
Positive Number: 23
Positive Number: 24
BUILD SUCCESSFUL (total time: 0 seconds)
```

5. Deduce a Java program with by creating three different threads (thread1 is going to create a sequence of even numbers from 2 to 50, thread2 is going to create a sequence of odd numbers from 1 to 49 and thread3 is going to create a sequence of prime numbers from 2 to 47)

```java
package lab6;
import java.util.*;

class R1 implements Runnable {
    public void run() {
       for(int i=2; i<=50; i++){
            if(i%2==0)
                System.out.println("Thread Running: Thread 1 - Even("+i+")");
            else
                continue;
       }
    }
}

class R2 implements Runnable {
    public void run() {
       for(int i=1; i<=49; i++){
            if(i%2!=0)
                System.out.println("Thread Running: Thread 2 - Odd("+i+")");
            else
                continue;
       }
    }
}

class R3 implements Runnable {
    public boolean prime(int num){
        if(num<2)
            return false;
        else{
            if(num==2) return true;
            else{
                for(int i=2; i<(int)((num/2)+1); i++){
                    if(num%i==0)
                        return false;
                }
                return true;
            }
        }
    }
    public void run() {
       for(int i=2; i<=47; i++){
            if(prime(i))
                System.out.println("Thread Running: Thread 3 - Prime("+i+")");
            else
                continue;
       }
    }
}

public class Lab6 {
    public static void main(String[] args) {
        Runnable r1 = new R1();
        Thread t1 = new Thread(r1);
        Runnable r2 = new R2();
        Thread t2 = new Thread(r2);
        Runnable r3 = new R3();
        Thread t3 = new Thread(r3);
        t1.start();
```

```
        t2.start();
        t3.start();
    }
}
```

OUTPUT:

```
Output - Lab6 (run)
run:
Thread Running: Thread 2 - Odd(1)
Thread Running: Thread 2 - Odd(3)
Thread Running: Thread 2 - Odd(5)
Thread Running: Thread 2 - Odd(7)
Thread Running: Thread 3 - Prime(2)
Thread Running: Thread 3 - Prime(3)
Thread Running: Thread 1 - Even(2)
Thread Running: Thread 3 - Prime(5)
Thread Running: Thread 2 - Odd(9)
Thread Running: Thread 3 - Prime(7)
Thread Running: Thread 1 - Even(4)
Thread Running: Thread 3 - Prime(11)
Thread Running: Thread 2 - Odd(11)
Thread Running: Thread 3 - Prime(13)
Thread Running: Thread 1 - Even(6)
Thread Running: Thread 3 - Prime(17)
Thread Running: Thread 2 - Odd(13)
Thread Running: Thread 3 - Prime(19)
Thread Running: Thread 2 - Odd(15)
Thread Running: Thread 1 - Even(8)
Thread Running: Thread 2 - Odd(17)
Thread Running: Thread 3 - Prime(23)
Thread Running: Thread 2 - Odd(19)
Thread Running: Thread 2 - Odd(21)
Thread Running: Thread 1 - Even(10)
Thread Running: Thread 2 - Odd(23)
Thread Running: Thread 3 - Prime(29)
Thread Running: Thread 2 - Odd(25)
Thread Running: Thread 1 - Even(12)
```

```
Output - Lab6 (run)
Thread Running: Thread 2 - Odd(27)
Thread Running: Thread 3 - Prime(31)
Thread Running: Thread 2 - Odd(29)
Thread Running: Thread 1 - Even(14)
Thread Running: Thread 2 - Odd(31)
Thread Running: Thread 3 - Prime(37)
Thread Running: Thread 3 - Prime(41)
Thread Running: Thread 2 - Odd(33)
Thread Running: Thread 1 - Even(16)
Thread Running: Thread 1 - Even(18)
Thread Running: Thread 1 - Even(20)
Thread Running: Thread 1 - Even(22)
Thread Running: Thread 1 - Even(24)
Thread Running: Thread 1 - Even(26)
Thread Running: Thread 1 - Even(28)
Thread Running: Thread 1 - Even(30)
Thread Running: Thread 2 - Odd(35)
Thread Running: Thread 2 - Odd(37)
Thread Running: Thread 2 - Odd(39)
Thread Running: Thread 3 - Prime(43)
Thread Running: Thread 3 - Prime(47)
Thread Running: Thread 2 - Odd(41)
Thread Running: Thread 1 - Even(32)
Thread Running: Thread 2 - Odd(43)
Thread Running: Thread 1 - Even(34)
Thread Running: Thread 2 - Odd(45)
Thread Running: Thread 2 - Odd(47)
Thread Running: Thread 2 - Odd(49)
Thread Running: Thread 1 - Even(36)
Thread Running: Thread 1 - Even(38)
```

```
Thread Running: Thread 1 - Even(40)
Thread Running: Thread 1 - Even(42)
Thread Running: Thread 1 - Even(44)
Thread Running: Thread 1 - Even(46)
Thread Running: Thread 1 - Even(48)
Thread Running: Thread 1 - Even(50)
BUILD SUCCESSFUL (total time: 0 seconds)
```