Practical 8: Multithreading

Name: Sutariya Savankumar

Roll no: MA065

1. **Write a Java program to do the following using threads created by extending Thread class,**
   - **Create 3 threads and display their properties as [thread-name, priority, group-name] – also display similar properties for main thread.**
   - **Override run() methods in the threads to display numbers from 1 to 5, 1 to 6 and 1 to 7 respectively.**
   - **Use following methods in the program as explained,**
     - **isAlive() – show whether the three threads are alive before and after using the start() method**
     - **sleep() – Use 1000, 2000 and 3000 milliseconds as the argument for the three threads respectively**
   - **Display messages to indicate start and end of the 3 threads as well as the main thread.**

Code

```
class ThreadDemo1 extends Thread {
    ThreadDemo1(String name) {
        super(name);
    }
    public void run() {
        System.out.println("Thread " + getName() + " started");
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread " + getName() + " : " + i);
            try {
                sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Thread " + getName() + " ended");
    }
}

class ThreadDemo2 extends Thread {
    ThreadDemo2(String name) {
        super(name);
    }
    public void run() {
```

```
        System.out.println("Thread " + getName() + " started");
        for (int i = 1; i <= 6; i++) {
            System.out.println("Thread " + getName() + " : " + i);
            try {
                sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Thread " + getName() + " ended");
    }
}

class ThreadDemo3 extends Thread {
    ThreadDemo3(String name) {
        super(name);
    }
    public void run() {
        System.out.println("Thread " + getName() + " started");
        for (int i = 1; i <= 7; i++) {
            System.out.println("Thread " + getName() + " : " + i);
            try {
                sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Thread " + getName() + " ended");
    }
}

public class p1 {
    public static void main(String[] args) {
        ThreadDemo1 t1 = new ThreadDemo1("Thread 1");
        ThreadDemo2 t2 = new ThreadDemo2("Thread 2");
        ThreadDemo3 t3 = new ThreadDemo3("Thread 3");
        System.out.println("Thread 1 is alive: " + t1.isAlive());
        System.out.println("Thread 2 is alive: " + t2.isAlive());
        System.out.println("Thread 3 is alive: " + t3.isAlive());
        System.out.println("Main thread is alive: " +
Thread.currentThread().isAlive());
        t1.start();
        t2.start();
        t3.start();
        System.out.println("Thread 1 is alive: " + t1.isAlive());
        System.out.println("Thread 2 is alive: " + t2.isAlive());
        System.out.println("Thread 3 is alive: " + t3.isAlive());
```

```
        System.out.println("Main thread is alive: " +
Thread.currentThread().isAlive());
        System.out.println("Main thread ended");
    }
}
```

Output

```
PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 8> java p1
Thread 1 is alive: false
Thread 2 is alive: false
Thread 3 is alive: false
Main thread is alive: true
Thread 1 is alive: true
Thread 2 is alive: true
Thread 3 is alive: true
Thread Thread 2 started
Thread Thread 2 : 1
Thread Thread 1 started
Main thread is alive: true
Thread Thread 3 started
Main thread ended
Thread Thread 1 : 1
Thread Thread 3 : 1
Thread Thread 1 : 2
Thread Thread 2 : 2
Thread Thread 1 : 3
Thread Thread 3 : 2
Thread Thread 1 : 4
Thread Thread 2 : 3
Thread Thread 1 : 5
Thread Thread 1 ended
Thread Thread 3 : 3
Thread Thread 2 : 4
Thread Thread 2 : 5
Thread Thread 3 : 4
Thread Thread 2 : 6
Thread Thread 3 : 5
Thread Thread 2 ended
Thread Thread 3 : 6
Thread Thread 3 : 7
Thread Thread 3 ended
```

2. **Write a Java Program for TrafficLight using Applet and Multithreading. Create three circle to display Red, Green and Yellow light which should simulate traffic light. The switching between Red, Green and Yellow should take place after every 10 seconds**

Code

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class p2 extends Applet implements Runnable
{
    Thread t;
    int i=0;
    public void init()
    {
        t=new Thread(this);
        t.start();
    }
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.fillOval(100,100,100,100);
        g.setColor(Color.green);
        g.fillOval(100,250,100,100);
        g.setColor(Color.yellow);
        g.fillOval(100,400,100,100);
        if(i==0)
        {
            g.setColor(Color.red);
            g.fillOval(100,100,100,100);
        }
        else if(i==1)
        {
            g.setColor(Color.green);
            g.fillOval(100,250,100,100);
        }
        else if(i==2)
        {
            g.setColor(Color.yellow);
            g.fillOval(100,400,100,100);
        }
    }
```

```
    public void run()
    {
        while(true)
        {
            try
            {
                Thread.sleep(10000);
                i++;
                if(i==3)
                    i=0;
                repaint();
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}
```

Output