

LAB ASSIGNMENTS:

SET A

A) Program to define a thread for printing text on output screen for 'n' number of times create 3 threds and run them. Pass the text 'n' parameters to the thresd constructor.

Examples:

I) First thread prints"COVID19" 10 times.

II) Second thread prints "LOCKDOWN2020" 20 TIMES

III) Third thread prints "VACCINATED2021" 30 times

```
public class A1 extends Thread
{
    String str;
    int n;

    A1(String str, int n)
    {
        this.str = str;
        this.n = n;
    }

    public void run()
    {
        try {
            for (int i = 0; i < n; i++) {
                System.out.println(getName() + " : " + str);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        A1 t1 = new A1("COVID19", 10);
        A1 t2 = new A1("LOCKDOWN2020", 20);
        A1 t3 = new A1("VACCINATED", 30);

        t1.start();
        t2.start();
        t3.start();

    }
}
```

b) Write a program in which thread sleep for 6 sec in the loop in reverse order from 100 to 1 and change the name of thread.

```
public class A2
{
```

```

        public static void main(String[] args)
    {
        try
        {
            Thread t = Thread.currentThread();
            t.setName("Reverse Thread");
            System.out.println(t);
            for (int i = 100; i >= 1; i--) {
                System.out.println(i);
                Thread.sleep(6000);
            }
        } // try
        catch (Exception e) {
            System.out.println(e);
        } // catch
    } // main
} // class

```

C) write a program to solve producer consumer problem in which a producer produces a value and consumer consume the value before producer generate the next value.

(Hint: use thread synchronization)

```

class shop {
    int material;
    boolean flag = false;

    public synchronized int get()
    {
        while (flag == false)
    {
        try
        {
            wait();
        } catch (Exception e)
        {
            e.printStackTrace();
        } // catch
    } // while
    flag = false;
    notify();
    return material;
} // get

    public synchronized void put(int value)
    {
        while (flag == true)
    {
        try
        {

```

```

        wait();
    } catch (Exception e)
    {
        e.printStackTrace();
    } // catch
} // while
material = value;
flag = true;
notify();
} // put

} // shop

class Consumer extends Thread {
    shop sh;
    int no;

    public Consumer(shop shp, int no) {
        sh = shp;
        this.no = no;
    } // consumerc

    public void run() {
        int value = 0;
        for (int i = 0; i < 10; i++) {
            value = sh.get();
            System.out.println("Consumer #" + this.no + " got: " + value);
        } // for
    } // run
} // Consumer

class Producer extends Thread {
    shop sh;
    int no;

    public Producer(shop s, int no) {
        sh = s;
        this.no = no;
    } // producerc

    public void run() {
        for (int i = 0; i < 10; i++) {
            sh.put(i);
            System.out.println("Producer #" + this.no + " put: " + i);
            try {
                sleep((int) (Math.random() * 1000));
            } catch (Exception e) {
                e.printStackTrace();
            } // catch
        } // for
    }
}

```

```

    }// run
} // Producer

public class A3 {
    public static void main(String[] args) {
        shop s = new shop();
        Producer p = new Producer(s, 1);
        Consumer c = new Consumer(s, 1);

        p.start();
        c.start();
    }
}

```

SET B:

a) Write a program to calculate the sum and average of an array of 1000 integers (generated randomly) using 10 threads. Each thread calculates the sum of 100 integers. Use these values to calculate average. [Use join method].but this program is for 10 integers you can make it according to your's need.

```

import java.util.*;

class Sum_Thread implements Runnable {
    Thread t;
    int a[] = new int[1000];
    int no, sum;

    Sum_Thread(String s, int n) {
        Random r = new Random();
        t = new Thread(this, s);
        int j = 0;
        no = n;
        for (int i = 0; i < 100; i++) {
            a[i] = r.nextInt(100);
            j++;
        }
        t.start();
    }

    } // Sum_Thread

    public void run() {
        try {
            for (int i = 0; i < 10; i++) {
                System.out.print(a[no] + " ");
                sum = sum + a[no];
            }
        }
    }
}

```

```

        no++;

    }
    System.out.println("");
    System.out.println("SUM: " + sum);
    System.out.println("AVG: " + sum / 10);
    System.out.println("");
} // try
catch (Exception e) {
    e.printStackTrace();
}

} // run

}

public class B1 {

    public static void main(String arg[]) {
        try {
            Sum_Thread t1 = new Sum_Thread("t1", 1);
            t1.t.join();

            Sum_Thread t2 = new Sum_Thread("t2", 10);
            t2.t.join();

            Sum_Thread t3 = new Sum_Thread("t3", 20);
            t3.t.join();

            Sum_Thread t4 = new Sum_Thread("t4", 30);
            t4.t.join();

            Sum_Thread t5 = new Sum_Thread("t5", 40);
            t5.t.join();

            Sum_Thread t6 = new Sum_Thread("t6", 50);
            t6.t.join();

            Sum_Thread t7 = new Sum_Thread("t7", 60);
            t7.t.join();

            Sum_Thread t8 = new Sum_Thread("t8", 70);
            t8.t.join();

            Sum_Thread t9 = new Sum_Thread("t9", 80);
            t9.t.join();

            Sum_Thread t10 = new Sum_Thread("t10", 90);
            t10.t.join();
        } // try
    }
}

```

```

        catch (Exception e) {
            e.printStackTrace();
        }

    }

}

```

B) Write a program for a simple search engine. Accept a string to be searched. Search for the string in all text files in the current folder. Use a separate thread for each file. The result should display the filename, line number where the string is found.

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.Scanner;

class Mythread extends Thread {
    String str;
    String filename;

    Mythread(String str, String filename) {
        this.str = str;
        this.filename = filename;
    }

    public void run() {
        try {
            int flag = 0;
            File f = new File(filename);
            BufferedReader br = new BufferedReader(new FileReader(f));
            String line = "";
            while ((line = br.readLine()) != null) {
                if (line.contains(str) == true) {
                    flag = 1;
                    break;
                }
            }
            if (flag == 1) {
                System.out.println("String found in folder/file :" + filename);
            } else {
                System.out.println("String not found in folder/file :" + filename);
            }
            br.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
}
}

public class B2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Search string :");
        String str = sc.nextLine();

        //Your folder name
        String dirname = "thread";
        File d = new File(dirname);
        if (d.isDirectory()) {
            String s[] = d.list();
            for (int i = 0; i < s.length; i++) {
                File f = new File(dirname + "/" + s[i]);
                if (f.isFile() && s[i].endsWith(".txt")) {
                    Mythread t = new Mythread(str, dirname + "/" + s[i]);
                    t.start();
                }
            }
        }
        sc.close();
    }
}

```

c) Write a program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

```

import java.util.*;

class FirstThread extends Thread {

    Random r = new Random();
    public int n;

    public void run() {
        try {
            for (int i = 0; i < 5; i++) {
                n = r.nextInt(10);
                System.out.println("Generated Random Number : " + n);

                if (n % 2 == 0) {
                    SecondThread t2 = new SecondThread(n);

```

```

        t2.start();
    } else {
        ThirdThread t3 = new ThirdThread(n);
        t3.start();
    }
    sleep(1000);
}
} // try
catch (Exception e) {
    e.printStackTrace();
}

} // run
} // class FirstThread

class SecondThread extends Thread {
    int square;

    SecondThread(int n) {
        square = n * n;
    }

    public void run() {
        try {
            System.out.println("Square OF Number: " + square);

        } // try
        catch (Exception e) {
            e.printStackTrace();
        }

    } // run
} // class second thread

class ThirdThread extends Thread {

    int cube;

    ThirdThread(int n) {
        cube = n * n * n;
    }

    public void run() {
        try {
            System.out.println("Cube OF Number: " + cube);

        } // try
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```



```

    }

    } // run

} // class second thread

public class B3 {

    public static void main(String arg[]) {

        FirstThread t1 = new FirstThread();

        t1.start();

    }

}

```

SET C:

A) Write a program that simulates a traffic light. The program lets the user select one of three lights: red,yellow. Or green with radio buttons. On selecting a button,an appropriates messages with “stop” or “ready” or “go” should appear above the buttons in a selected color. Initially there is no message shown.

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class C1 extends JFrame implements ActionListener {
    JPanel panel = new JPanel();

    JLabel title_lbl = new JLabel("Traffic Light");

    JRadioButton rb1 = new JRadioButton("Red");
    JRadioButton rb2 = new JRadioButton("Yellow");
    JRadioButton rb3 = new JRadioButton("Green");

    JButton ok_btn = new JButton("OK");

    ButtonGroup g1 = new ButtonGroup();

    JLabel red;
    JLabel yellow;
    JLabel green;

    C1() {
        setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        setSize(800, 700);
    }
}

```

```
setTitle("Java Project Program");

setLocation(400, 150);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

red = new JLabel("STOP");
red.setForeground(Color.red);
yellow = new JLabel("READY");
yellow.setForeground(Color.yellow);
green = new JLabel("GO");
green.setForeground(Color.green);

panel.setPreferredSize(new Dimension(500, 400));
panel.setLayout(null);
panel.setBackground(Color.lightGray);

title_lbl.setBounds(200, 0, 200, 100);
panel.add(title_lbl);

rb1.setBounds(100, 210, 100, 20);
panel.add(rb1);
rb2.setBounds(200, 210, 100, 20);
panel.add(rb2);
rb3.setBounds(300, 210, 100, 20);
panel.add(rb3);

ok_btn.setBounds(200, 310, 100, 50);
panel.add(ok_btn);
ok_btn.addActionListener(this);

red.setBounds(220, 110, 100, 50);
yellow.setBounds(220, 110, 100, 50);
green.setBounds(220, 110, 100, 50);

rb1.setForeground(Color.red);
rb2.setForeground(Color.yellow);
rb3.setForeground(Color.green);

rb1.setBackground(Color.lightGray);
rb2.setBackground(Color.lightGray);
rb3.setBackground(Color.lightGray);

red.setFont(new Font("Arial", Font.BOLD, 20));
yellow.setFont(new Font("Arial", Font.BOLD, 20));
green.setFont(new Font("Arial", Font.BOLD, 20));
title_lbl.setFont(new Font("Arial", Font.BOLD, 20));

panel.add(red);
panel.add(yellow);
panel.add(green);
```

```

        red.setVisible(false);
        yellow.setVisible(false);
        green.setVisible(false);

        g1.add(rb1);
        g1.add(rb2);
        g1.add(rb3);

        this.add(panel);
        this.pack();
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == ok_btn) {
            if (rb1.isSelected()) {
                yellow.setVisible(false);
                green.setVisible(false);
                red.setVisible(true);
            } else if (rb2.isSelected()) {
                red.setVisible(false);
                green.setVisible(false);
                yellow.setVisible(true);
            }
            if (rb3.isSelected()) {
                red.setVisible(false);
                yellow.setVisible(false);
                green.setVisible(true);
            }
        }
    }

    public static void main(String args[]) {
        new C1();
    }
}

```