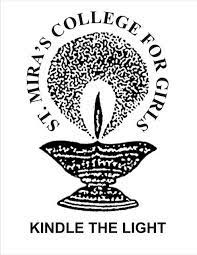
**SADHU VASWANI MISSION’S**

**ST. MIRA’S COLLEGE FOR GIRLS**

**6, Koregaon Road, Pune – 411 001.**

**BACHELOR OF BUSINESS ADMINISTRATION (COMPUTER APPLICATION)**

**DEPARTMENT OF COMMERCE**

***CERTIFICATE***

***This is to certify that Ms.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***of T.Y.B.B.A.(CA) Seat No. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_has successfully completed\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Practicals / Assignment for Lab course in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_as laid down by the Savitribai Phule Pune University for the Academic Year\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_***

**Subject Teacher Course Coordinator Principal**

**Pune**

**Date:**

**Practical Examinations**

**Exam Seat No. : Date :**

**Internal Examiner External Examiner**

**Q1. Write a java program to display IP Address and Name of client machine.**

// Java program to find IP address of your computer

// java.net.InetAddress class provides method to get

// IP of any host name

import java.net.\*;

import java.io.\*;

import java.util.\*;

import java.net.InetAddress;

public class JavaProgram

{

public static void main(String args[]) throws Exception

{

// Returns the instance of InetAddress containing

// local host name and address

InetAddress localhost = InetAddress.getLocalHost();

System.out.println("System IP Address : " +

(localhost.getHostAddress()).trim());

// Find public IP address

String systemipaddress = "";

try

{

URL url\_name = new URL("http://bot.whatismyipaddress.com");

BufferedReader sc =

new BufferedReader(new InputStreamReader(url\_name.openStream()));

// reads system IPAddress

systemipaddress = sc.readLine().trim();

}

catch (Exception e)

{

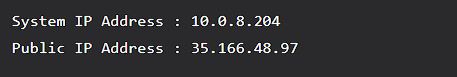
systemipaddress = "Cannot Execute Properly";

}

System.out.println("Public IP Address: " + systemipaddress +"\n");

}

}



**Q3. Write a multithreading program in java to display all the vowels from a given String. (Use Thread Class)**

import java.lang.\*;

import java.util.\*;

class Vowels extends Thread

{

String s1;

Vowels(String s)

{ s1=s;

start();

}

public void run()

{

System.out.println("Vowels are ");

for(int i=0;i<s1.length();i++)

{

char ch=s1.charAt(i);

if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')

System.out.print(" "+ch);

}

}

}

public class Demo1 {

public static void main(String[] args)

{

Scanner sn=new Scanner(System.in);

System.out.println("Enter a string");

String str1=sn.next();

Vowels v=new Vowels(str1);

}

}



**Q6. Write a java program to display “Hello Java” message n times on the screen. (Use Runnable Interface).**

class mythread implements Runnable

{

Thread t;

public mythread(String title)

{

t=new Thread(this,title);

t.start();

}

public void run()

{

for(int i=0;i<n;i++)

{

System.out.println((i+1)+"ThreadName:"+Thread.currentThread().getName());

try

{

Thread.sleep(100);

}

catch(Exception e)

{

}

}}}

public class roll no 16

{

public static void main(String a [])

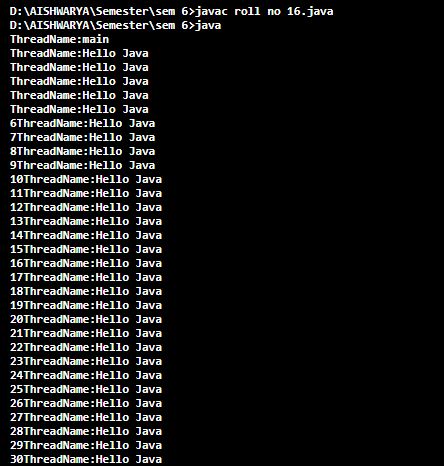
{

System.out.println("ThreadName:"+Thread.currentThread().getName());

mythread mt=new mythread("Hello Java");

}

}



**Q25. Write a java program which will display name and priority of current thread. Change name of Thread to MyThread and priority to 2. Display the details of Thread.**

public class MainThread

{

public static void main(String arg[])

{

Thread t=Thread.currentThread();

System.out.println("Current Thread:"+t);

//Change Name t.setName("My Thread ");

System.out.println ("After the name is Changed:"+t);

try {

for(int i=2;i>0;i--)

{

System.out.println(i);

Thread.sleep(1000);

}

}

catch(Exception e)

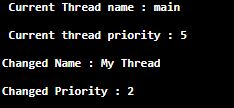
{

System.out.println(e);

}

}

}



**Q26.** **Write a java program using multithreading to execute the threads sequentially.**

**(Use Synchronized Method)**

class multiT extends Thread

{

multiT1 t;

String name;

public multiT(String s, multiT1 t1)

{

name=s;

start();

t=new multiT1();

t1=t;

}

public void run()

{

t.Display(name);

}

}

class multiT1

{

synchronized void Display(String name1)

{

try

{

for(int i=1; i<=5; i++)

{

System.out.println(name1+" "+i);

}

}catch(Exception e)

{}

}

}

public class adprgm26

{

public static void main(String[] args)

{

multiT1 td=new multiT1();

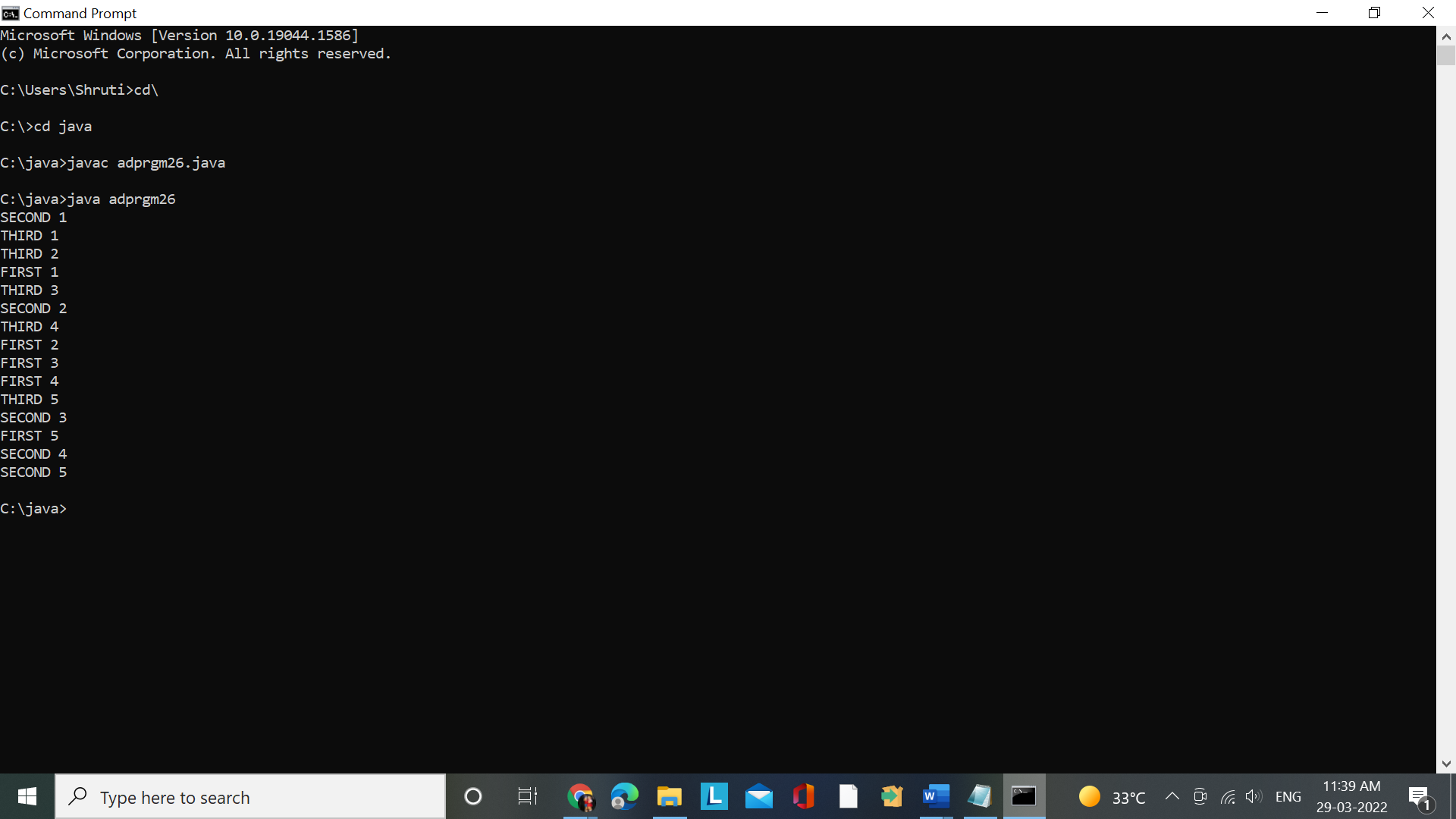
multiT d=new multiT("FIRST",td);

multiT d1=new multiT("SECOND",td);

multiT d2=new multiT("THIRD",td);

}

}



**Q40. Write a Multithreading program in java to display all the alphabets from A to Z after 3 seconds.**

public class roll no 16\_1 extends Thread

{

char c;

public void run()

{

for(c = 'A'; c<='Z';c++)

{

System.out.println(""+c);

try

{

Thread.sleep(3000);

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

public static void main(String args[])

{

Roll no 16\_1 t = new Slip40\_1();

t.start();

}

}



**Q24. Write a program in java which will show lifecycle (creation, sleep, and dead) of a thread. Program should print randomly the name of thread and value of sleep time. The name of the thread should be hard coded through constructor. The sleep time of a thread will be a random integer in the range 0 to 4999.**

class MyThread extends Thread{

public MyThread(String s){

super(s);

}

public void run(){

System.out.println(getName()+" thread created.");

while(true){

System.out.println(this);

int s = (int)(Math.random()\*5000);

System.out.println(getName()+" is sleeping for "+s+"msec");

try{

Thread.sleep(s);

}catch(Exception e){}

}

}

}

class ThreadLifeCycle{

public static void main(String args[]){

MyThread t1 = new MyThread("A"),

t2 = new MyThread("B”);

t1.start();

t2.start();

try{ t1.join();

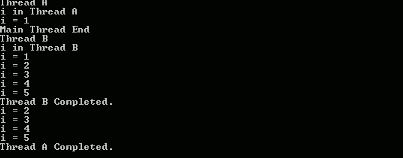
t2.join();

}catch(Exception e){}

System.out.println(t1.getName()+" thread end.");

System.out.println(t2.getName()+" thread end.");

}

}

**Q29. Write a Multithreading program in java to convert smile face into the crying face after 5**

**seconds and vice versa (Use Applet).**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code= "smileface.java" height="600" width="600">

</applet>

\*/

public class smileface extends java.applet.Applet implements Runnable {

int aflag;

Thread t;

public void init() {

t=new Thread(this); aflag=0;

t.start();

}

public void run()

{

try

{

if (aflag==0)

{

t.sleep(1000);

aflag=1;

} else

{

t.sleep(1000);

aflag=0;

}

repaint();

run();

}

catch(Exception e)

{

}

}

public void paint(Graphics g) {

g.drawOval(100,100,100,100);

g.fillOval(120,125,20,20);

g.fillOval(160,125,20,20);

g.drawLine(150,135,150,165);

if (aflag==0)

{ g.drawArc(140,160,20,20,0,-180);

aflag=1;

}

else

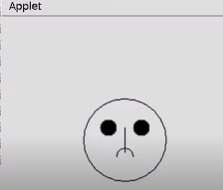
{

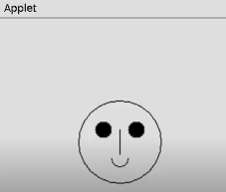
g.drawArc(140,160,20,20,0,180);

aflag=0;

}

}

 }



**Q45. Write applet, called Checkers, a red oval (a checker piece) moves from a black**

**square to a white square, as if on a checkerboard**

import java.awt.Graphics;

import java.awt.Color;

public class Checkers extends java.applet.Applet

implements Runnable {

Thread runner;

int xpos;

public void start() {

if (runner == null) {

runner = new Thread(this);

runner.start();

}

}

public void stop() {

if (runner != null) {

runner.stop();

runner = null;

}

}

public void run() {

setBackground(Color.blue);

while (true) {

for (xpos = 5; xpos <= 105; xpos+=4) {

repaint();

try { Thread.sleep(100); }

catch (InterruptedException e) { }

}

xpos = 5;

}

}

public void paint(Graphics g) {

// Draw background

g.setColor(Color.black);

g.fillRect(0, 0, 100, 100);

g.setColor(Color.white);

g.fillRect(101, 0, 100, 100);

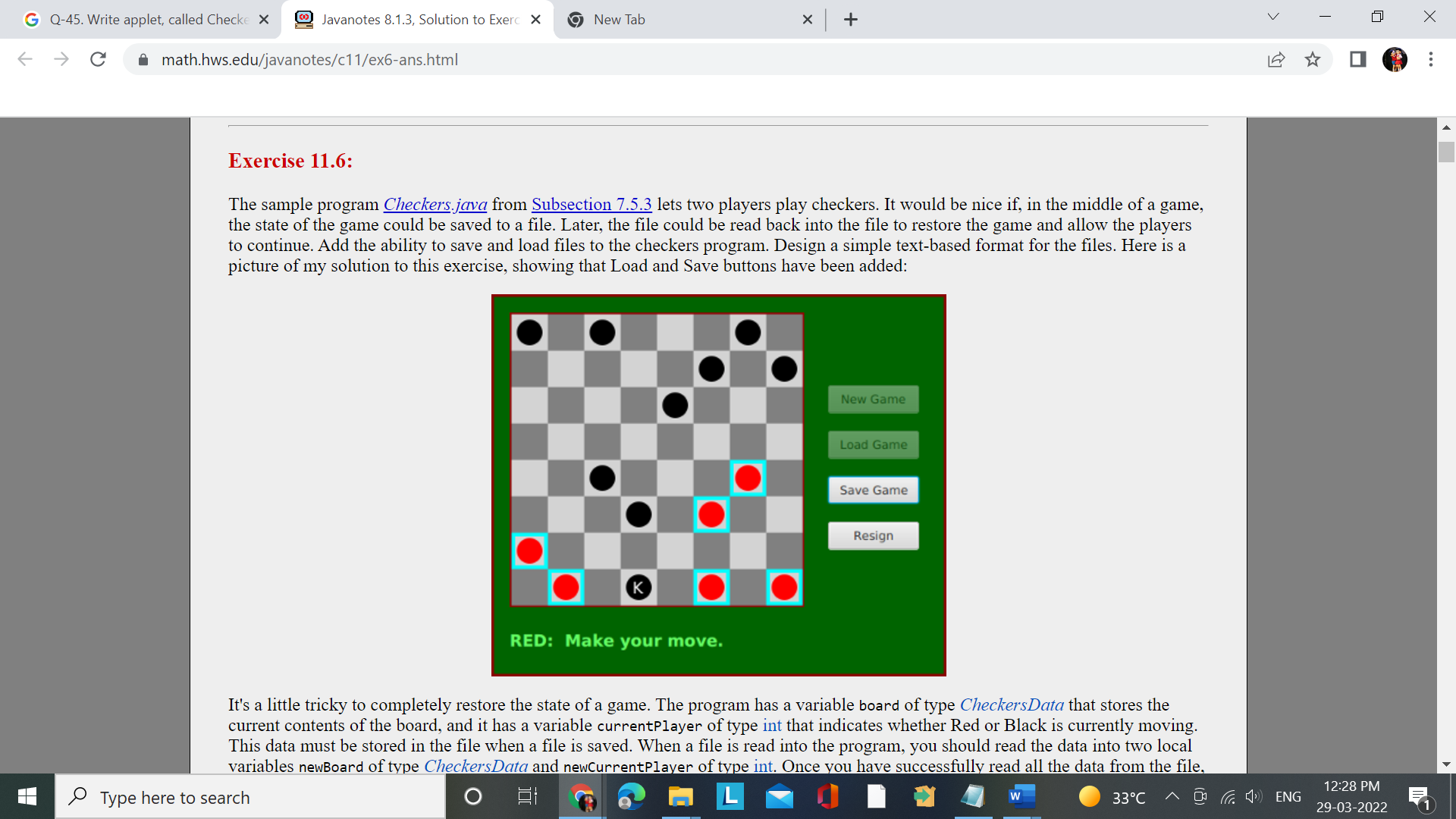
// Draw checker

g.setColor(Color.red);

g.fillOval(xpos, 5, 90, 90);

}

}



**Q47. Write the ColorSwirl applet prints a single string to the screen ("All the Swirly Colors"), but that string is presented in different colors that fade into each other dynamically .**

import java.awt.Graphics;

import java.awt.Color;

import java.awt.Font;

public class ColorSwirl extends java.applet.Applet

implements Runnable {

Font f = new Font("TimesRoman",Font.BOLD,48);

Color colors[] = new Color[50];

Thread runThread;

public void start() {

if (runThread == null) {

runThread = new Thread(this);

runThread.start();

}

}

public void stop() {

if (runThread != null) {

runThread.stop();

runThread = null;

}

}

public void run() {

// initialize the color array

float c = 0;

for (int i = 0; i &lt; colors.length; i++) {

colors[i] =

Color.getHSBColor(c, (float)1.0,(float)1.0);

c += .02;

}

// cycle through the colors

int i = 0;

while (true) {

setForeground(colors[i]);

repaint();

i++;

try { Thread.sleep(50); }

catch (InterruptedException e) { }

if (i == colors.length ) i = 0;

}

}

public void paint(Graphics g) {

g.setFont(f);

g.drawString("All the Swirly Colors", 15, 50);

}

}

import java.awt.Graphics;

2: import java.awt.Color;

3: import java.awt.Font;

4:

continues

184

Simple Animation and Threads

M T W R F S S

DAY

10

5: public class ColorSwirl extends java.applet.Applet

6: implements Runnable {

7:

8: Font f = new Font(“TimesRoman”,Font.BOLD,48);

9: Color colors[] = new Color[50];

10: Thread runThread;

11:

12: public void start() {

13: if (runThread == null) {

14: runThread = new Thread(this);

15: runThread.start();

16: }

17: }

18:

19: public void stop() {

20: if (runThread != null) {

21: runThread.stop();

22: runThread = null;

23: }

24: }

25:

26: public void run() {

27:

28: // initialize the color array

29: float c = 0;

30: for (int i = 0; i < colors.length; i++) {

31: colors[i] =

32: Color.getHSBColor(c, (float)1.0,(float)1.0);

33: c += .02;

34: }

35:

36: // cycle through the colors

37: int i = 0;

38: while (true) {

39: setForeground(colors[i]);

40: repaint();

41: i++;

42: try { Thread.sleep(50); }

43: catch (InterruptedException e) { }

44: if (i == colors.length ) i = 0;

45: }

46: }

47:

48: public void paint(Graphics g) {

49: g.setFont(f);

50: g.drawString(“All the Swirly Colors”, 15,50);

51: }

52: }

**Q13. Write a Multithreading program using Runnable interface to blink Text on the frame.**

import java.awt.\*;

import java.awt.event.\*;

public class BlinkText extends Frame implements Runnable

{

Thread t;

Label l1;

int f;

public BlinkText()

{

t=new Thread(this);

t.start();

setLayout(null);

l1=new Label("Hi");

l1.setBounds(100,100,100,40);

add(l1);

setSize(300,300);

setVisible(true); f=0;

}

public void run()

{

try

{

if(f==0)

{

t.sleep(200);

l1.setText("");

f=1;

}

if(f==1)

{

t.sleep(200);

l1.setText("Hello Java");

f=0;

}

}catch(Exception e)

{

System.out.println(e);

}

run();

}

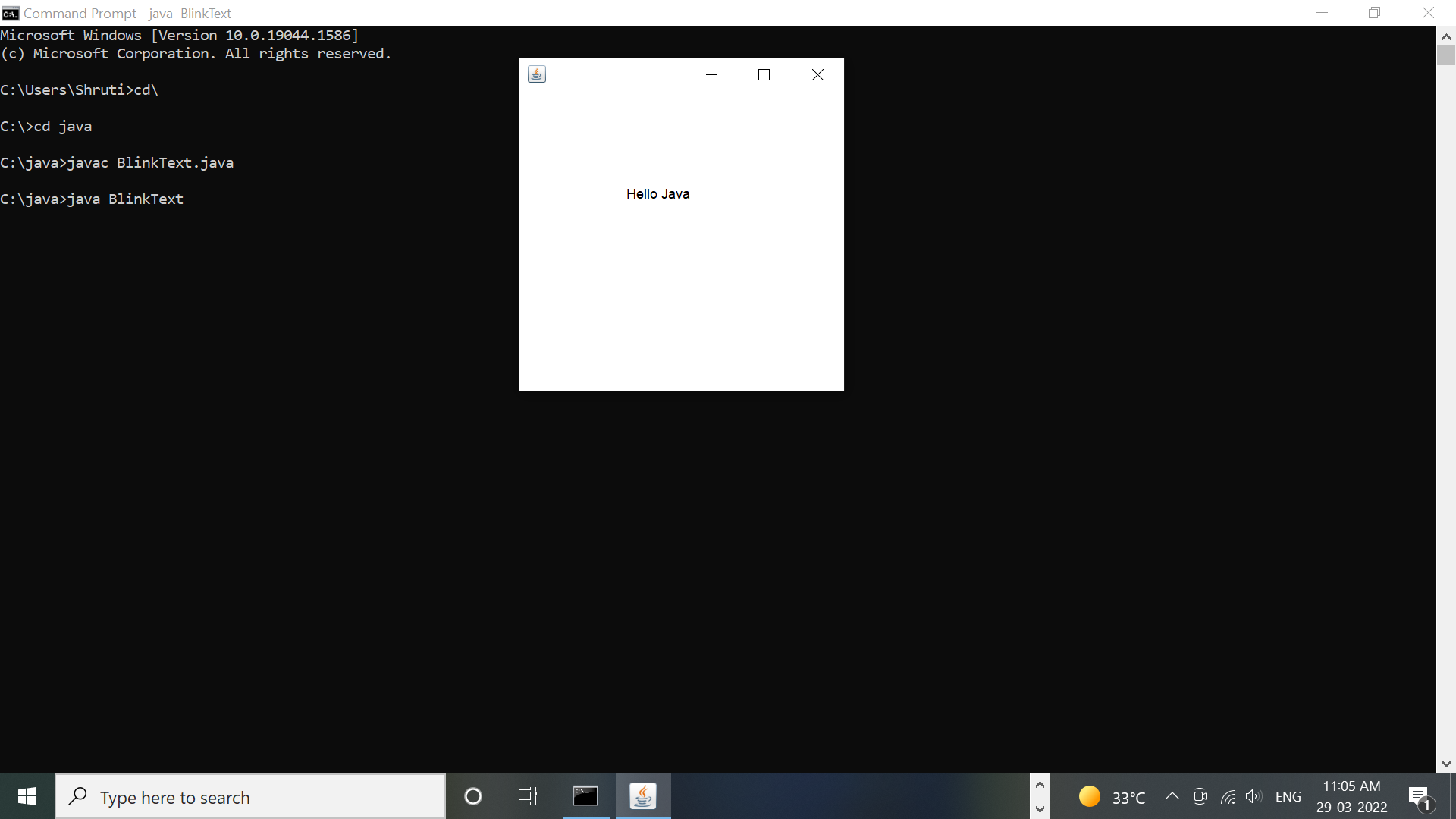
public static void main(String args[])

{

new BlinkText();

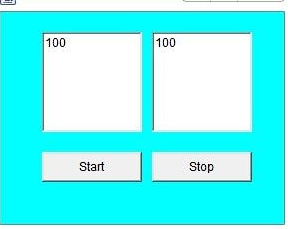
}

}



**Q18. Write a Multithreading program in java to display the number’s between 1 to 100 continuously in a TextField by clicking on button. (use Runnable Interface).**

import java.awt.\*;  
import java.awt.event.\*;  
class  ThreadClass extends Frame implements ActionListener,Runnable  
{  
     Button b1, b2;  
     TextField txt1, txt2;  
     int cnt;  
     Thread t1 = new Thread(this, "txt1");  
     Thread t2 = new Thread(this, "txt12");  
     public ThreadClass()  
     {  
          setLayout(null);  
          txt1 = new TextField();  
          txt2 = new TextField();  
          b1 = new Button("Start");  
          b2 = new Button("Stop");  
          txt1.setBounds(50,50,100,100);  
          txt2.setBounds(160,50,100,100);  
          b1.setBounds(50,170,100,30);  
          b2.setBounds(160,170,100,30);  
          add(txt1);  
          add(txt2);  
          b1.addActionListener(this);  
          b2.addActionListener(this);  
          add(b1);  
          add(b2);  
          setSize(400,400);  
          setVisible(true);  
          cnt=0;  
          addWindowListener(new WindowAdapter()  
          {  
               public void windowClosing(WindowEvent e)  
               {  
                    System.exit(0);  
               }  
          });  
     }  
     public void actionPerformed(ActionEvent ae)  
     {  
          String str;  
          str=ae.getActionCommand();  
          if (str.equals("Start"))  
          {  
               t1.start();  
               t2.start();  
          }  
          else if (str.equals("Stop"))  
          {  
               t1.stop();  
               t2.stop();  
          }  
     }  
     public void run()  
     {  
          try  
          {  
               for (int i=1; i<=100;i++)  
               {  
                    txt1.setText(""+i);  
                    t1.sleep(150);  
                    txt2.setText(""+i);  
                    t2.sleep(150);  
               }  
          }  
          catch (Exception ex)  
          {  
               ex.printStackTrace();  
          }  
     }  
     public static void main(String[] args)  
     {  
          new ThreadClass().show();  
     }  
}



**Q43. Write a Multithreading program in java to create an applet that contains a TextField to show time. The time should be displayed in the hh:mm:ss format. The thread should start when the user clicks the Start button and stop when the user clicks the stop button. Initialize the values to current time.**

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

public class rollno16 extends Frame implements

ActionListener,Runnable

{

Button start,stop;

TextField tf;

int x=0,y=0;

String msg="";

Thread t1=new Thread(this);

public slip27()

{

setLayout(new FlowLayout());

start=new Button("start");

stop=new Button("stop"); add(start);

add(stop);

start.addActionListener(this);

stop.addActionListener(this);

addWindowListener(new

WindowAdapter()

{

public void

windowClosing(WindowEvent e)

{

System.exit(0);

}

});

setSize(200,200);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

Button btn=(Button)ae.getSource();

if(btn==start)

{

t1.start();

}

if(btn==stop)

{

t1.stop();

}

}

public void run()

{

try

{ while(true)

{

repaint();

Thread.sleep(350);

}

}

catch(Exception e)

{

}

}

public void paint(Graphics g)

{

int sec,min,hr;

GregorianCalendar date = new GregorianCalendar();

sec = date.get(Calendar.SECOND);

min = date.get(Calendar.MINUTE);

hr = date.get(Calendar.HOUR);

msg = hr+":"+min+":"+sec;

g.drawString(msg,10,y+=10);

}

public static void main(String args[])

{

new rollno16();

}

} 

**Q5. Write a java program to simulate traffic signal using multithreading.**

import java.applet.\*;

import java.awt.\*;

class Slip3\_2 extends Applet implements Runnable

{

Thread t;

int r,g1,y,i;

public void init()

{

T=new Thread(this);

t.start();

r=0; g1=0;I=0; y=0;

}

public void run()

{

try

{

for(I =24; I >=1;i--)

{

if (I >16&& I <=24)

{

t.sleep(200);

r=1;

repaint();

}

if (I >8&& I <=16)

{

t.sleep(200);

y=1;

repaint();

}

if(I >1&& I <=8)

{

t.sleep(200);

g1=1;

repaint();

}

}

if (I ==0)

{

run();

}

}

catch(Exception e)

{ System.out.println(e);

}

} public void paint(Graphics g)

{

g.drawRect(100,100,100,300);

if (r==1)

{

g.setColor(Color.red);

g.fillOval(100,100,100,100);

g.setColor(Color.black);

g.drawOval(100,200,100,100);

g.drawOval(100,300,100,100);

r=0;

}

if (y==1)

{

g.setColor(Color.black);

g.drawOval(100,100,100,100);

g.drawOval(100,300,100,100);

g.setColor(Color.yellow);

g.fillOval(100,200,100,100);

y=0;

}

if (g1==1)

{

g.setColor(Color.black);

g.drawOval(100,100,100,100);

g.drawOval(100,200,100,100);

g.setColor(Color.green);

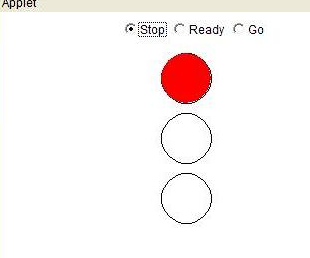
g.fillOval(100,300,100,100);

g1=0;

}

}

}



**Q12. Write a Multithreading program in java for Racing Car. (Use AWT)**

import java.awt.\*;  
import java.util.\*;  
import java.applet.\*;  
//The basic applet class.The applet shows 4 cars crossing each other at a square.  
class Slip7\_2 extends Applet implements Runnable  
{  
Thread t;  
//4 variables used to vary the car's positions.  
int x1=0,x2=380,y1=50,y2=250;  
public void start()  
{  
if(t==null)  
{  
t =new Thread(this,"New Thread");//New side Thread created on start  
of applet.  
t.start();  
}  
}  
public void stop()  
{  
if(t!=null)  
{  
t =null;//On stop of applet the created thread is destroyed.  
}  
}  
//Implementation of method run() of Runnable interface.  
public void run()  
{  
Thread t1=Thread.currentThread();  
while(t==t1)  
{  
repaint();  
try  
{  
Thread.sleep(100);  
}  
catch(Exception e)  
{ }  
}  
}  
public void paint(Graphics g)  
{  
setBackground(Color.cyan);  
g.setColor(Color.BLACK);  
x1=(x1+16)%400;  
x2=x2-16;  
y1=(y1+12)%300;  
y2=y2-12;  
if(y2<0)  
y2=288;  
if(x2<0)  
x2=384;  
//Draw the roads using 2 filled rectangles using black color.  
g.fillRect(0,130,400,40);  
g.fillRect(180,0,40,305);  
//Draw the white colored lines.  
g.setColor(Color.white);  
for(int i =0; i <20;i++)  
{  
if(i !=9 &&i !=10)  
g.drawLine(i\*20,150,i\*20+10,150);  
}  
  
for(int j=0;j<15;j++)  
{  
if(j!=7 && j!=8)  
g.drawLine(200,j\*20,200,j\*20+10);  
}  
//Draw 4 colored cars using filled round rectangles.  
g.setColor(Color.red);  
g.fillRoundRect(x2,152,20,8,2,2);  
g.fillRoundRect(x1,140,20,8,2,2);  
g.fillRoundRect(190,y1,8,20,2,2);  
g.fillRoundRect(202,y2,8,20,2,2);  
}  
}

**Q9. Write a MultiThreading program in java using Runnable interface to draw temple flag on an applet container.**

import java.awt.\*;

import java.applet.\*;

/\* <APPLET code= "flag.class" width= "500" height= "300">

</APPLET> \*/

public class flag extends Applet implements Runnable

{

Thread t;

int x1,x2,x3,y3,x4,y4,x5,ln;

public void init()

{

t=new Thread(this);

t.start();

ln=1;

}

public void run()

{

try{ if(ln==1) { for(x1=200;x1>100;)

{

t.sleep(200);

repaint();

}

}

ln=2;

if(ln==2) { for(x2=100;x2<150;)

{

t.sleep(200);

repaint();

}

}

ln=3;

if(ln==3) { for(x3=150,y3=100;x3>125&&y3<125;)

{

t.sleep(200);

repaint();

}

}

ln=4;

if(ln==4) { for(x4=125,y4=125;x4<150&&y4<150;)

{

t.sleep(200);

repaint();

}

}

ln=5;

if(ln==5) { for(x5=150;x5>100;)

{

t.sleep(200);

repaint();

}

}

ln=1;

}catch(Exception e){

System.out.println(e);

}

run();

}

public void paint(Graphics g)

{

if(ln==1&&x1>100)

{

g.drawLine(100,200,100,x1-=5);

}

if(ln==2&&x2<150)

{

g.drawLine(100,200,100,100);

g.drawLine(100,100,x2+=5,100);

}

if(ln==3&&x3>125&&y3<125)

{

g.drawLine(100,200,100,100);

g.drawLine(100,100,150,100);

g.drawLine(150,100,x3-=5,y3+=5);

}

if(ln==4&&x4<150&&y4<150)

{

g.drawLine(100,200,100,100);

g.drawLine(100,100,150,100);

g.drawLine(150,100,125,125);

g.drawLine(125,125,x4+=5,y4+=5);

}

if(ln==5&&x5>100)

{

g.drawLine(100,200,100,100);

g.drawLine(100,100,150,100);

g.drawLine(150,100,125,125);

g.drawLine(125,125,150,150);

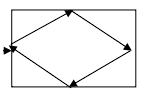
g.drawLine(150,150,x5-=5,150);

}

}

}

**Q15. Write a Multithreading program in java using Runnable interface to move text on the frame as follow: Starting Position of Text**



import java.awt.\*;

import java.awt.event.\*;

class Slip9\_2 extends Frame implements Runnable

{ Label l1;

Thread t;

int x,y,side;

Slip9\_2()

{

setLayout(null);

l1=new Label(" Hello Java");

l1.setFont(new Font("",Font.BOLD,14));

l1.setForeground(Color.red);

setSize(400,400);

setVisible(true);

t=new Thread(this);

t.start();

x=5; y=200; side=1;

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{ System.exit(0);

}

});

}

public void run()

{

try

{

if(side==1)

{ t.sleep(50);

l1.setBounds(x+=5,y-=5,80,15);

add(l1);

if(y==20)

side=2;

}

if(side==2)

{ t.sleep(50);

l1.setBounds(x+=5,y+=5,80,15);

add(l1);

if(y==200)

side=3;

}

if(side==3)

{ t.sleep(50);

l1.setBounds(x-=5,y+=5,80,15);

add(l1);

if(y==390)

side=4;

}

if(side==4)

{ t.sleep(50);

l1.setBounds(x-=5,y-=5,80,15);

add(l1);

if(x==0)

{ side=1; x=0; y=200;

}

}

}

catch(Exception e)

{

System.out.println(e);

}

run();

}

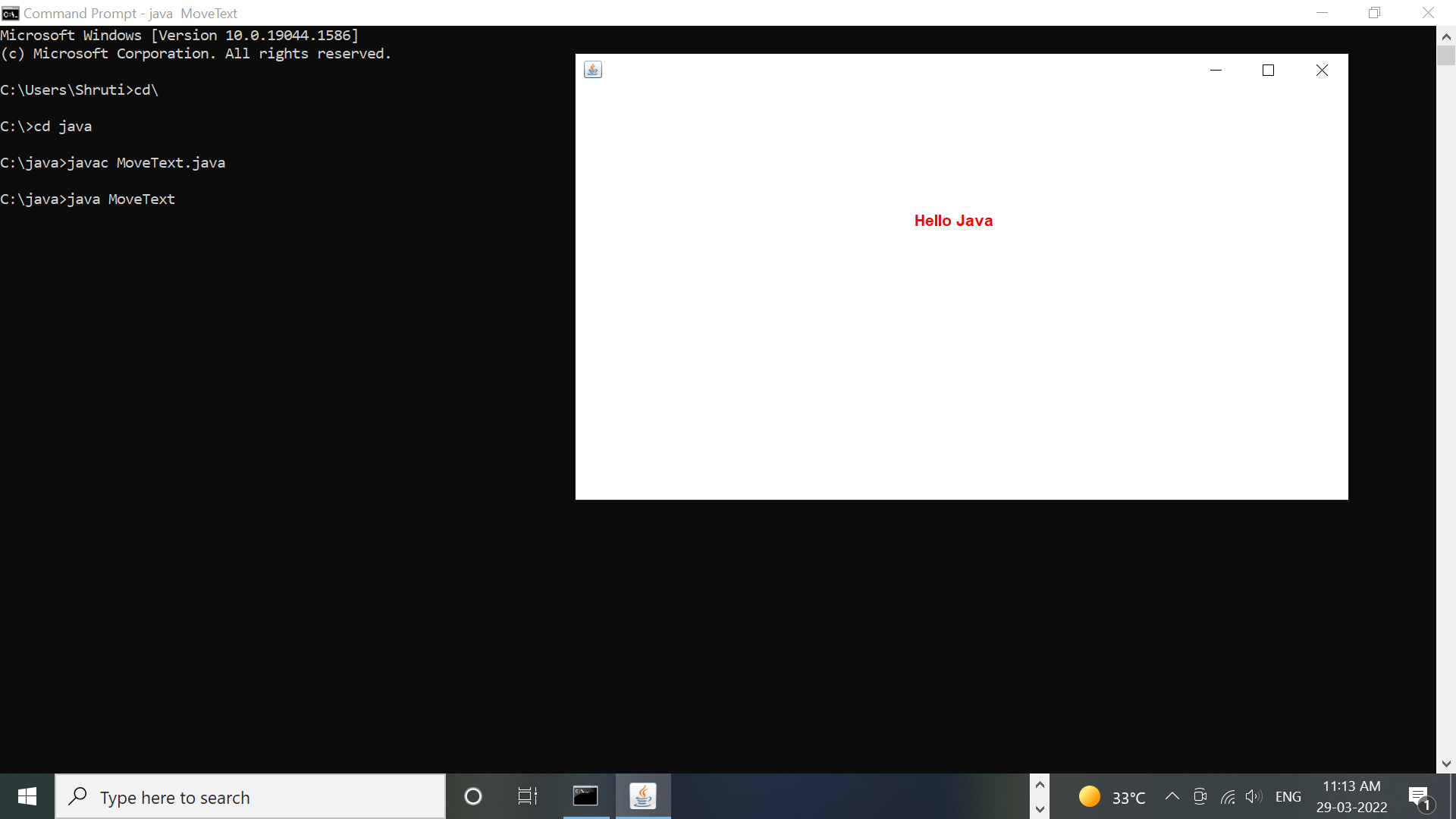
public static void main(String args[])

{

new Slip9\_2();

}

}



**Q16. Write a Multithreading program in java for bouncing ball. For each bounce, Change the color of ball randomly.**

import java.awt.\*;

/\*<applet code="BounsingBall.class" height=400 width=350></applet>\*/

public class BounsingBall extends java.applet.Applet implements Runnable

{

Thread t;

int f,y,f1,f2,f3;

public void init()

{

t=new Thread(this);

t.start();

f=0; y=0; f1=0;

}

public void run()

{ try{

if (f==0){ t.sleep(10);

y=y+5;

repaint();

if(f1==6)

f1=0;

}

if(f==1) { t.sleep(10);

y=y-5;

repaint();

if(f1==6)

f1=0;

}

}catch(Exception e){ }

run();

}

public void paint(Graphics g)

{

if(f==0) {

if(f1==1)

g.setColor(Color.green);

if(f1==2)

g.setColor(Color.blue);

if(f1==3)

g.setColor(Color.red);

if(f1==4)

g.setColor(Color.yellow);

if(f1==5)

g.setColor(Color.orange);

g.fillOval(150,y+10,20,20);

if(y==400)

{

f1++; f=1;

}

}

if(f==1) {

if(f1==1)

g.setColor(Color.green);

if(f1==2)

g.setColor(Color.blue);

if(f1==3)

g.setColor(Color.red);

if(f1==4)

g.setColor(Color.yellow);

if(f1==5)

g.setColor(Color.orange);

g.fillOval(150,y-10,20,20);

if(y==0)

{

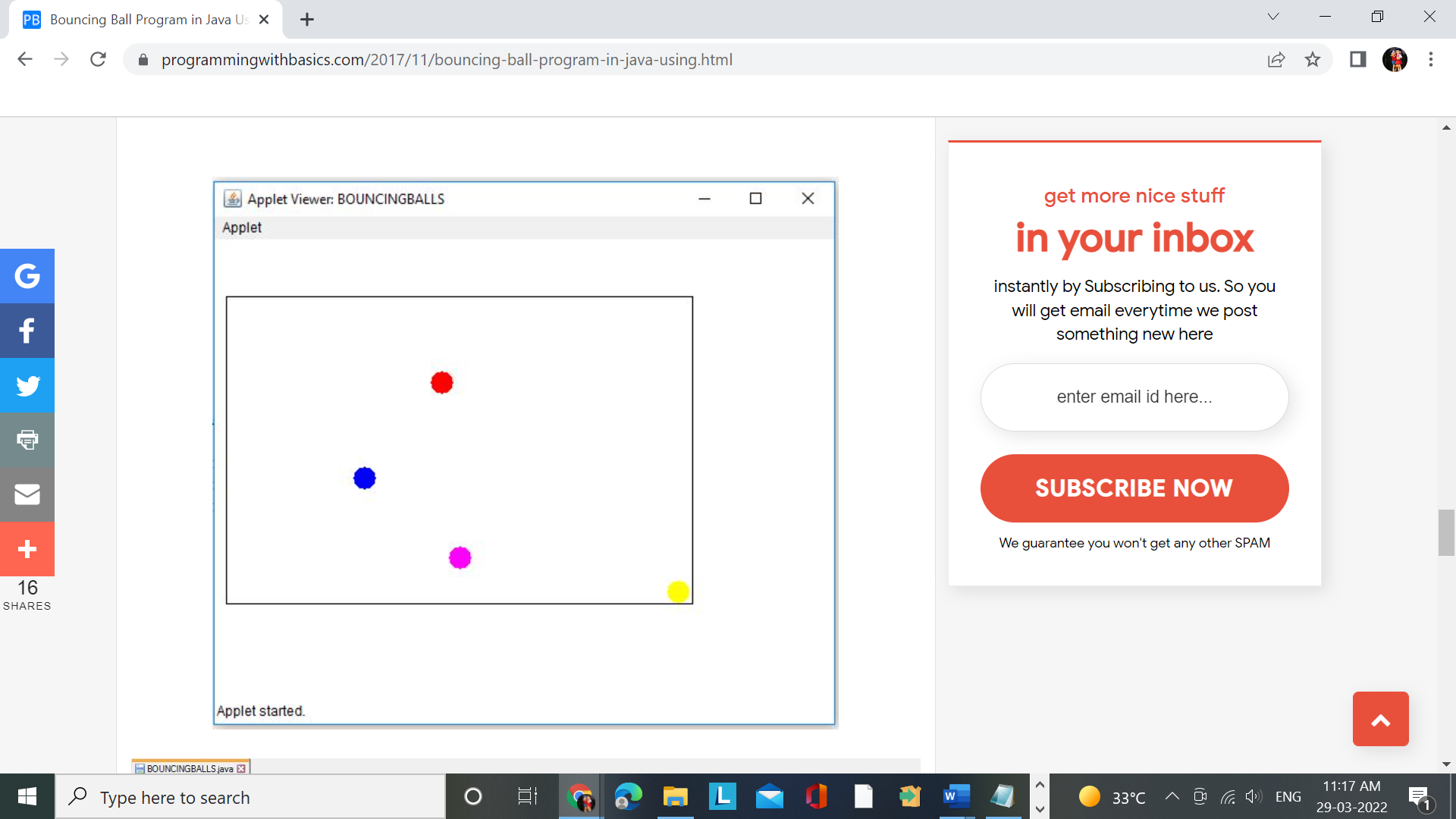
f1++; f=0;

}

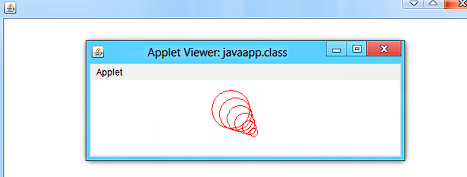
}

}

}



**Q37. Write an applet programme to create a oval animation starting from small oval and after every time instance one size bigger oval should be drawn making a shape of cone.**



/\*

Draw Oval & Circle in Applet Window Example

This java example shows how to draw ovals & circles in an applet window using

drawOval method of Graphics class. It also shows how to draw a filled

ovals and circles using fillOval method of Graphics class.

\*/

/\*

<applet code="DrawOvalsExample" width=500 height=500>

</applet>

\*/

import java.applet.Applet;

import java.awt.Color;

import java.awt.Graphics;

public class DrawOvalsExample extends Applet{

public void paint(Graphics g){

//set color to red

setForeground(Color.red);

/\*

\* to draw a oval in an applet window use,

\* void drawOval(int x1,int y1, int width, int height)

\* method.

\*

\* This method draws a oval of specified width and

\* height at (x1,y1)

\*/

//this will draw a oval of width 50 & height 100 at (10,10)

g.drawOval(10,10,50,100);

/\*

\* To draw a filled oval use

\* fillOval(int x1,int y1, int width, int height)

\* method of Graphics class.

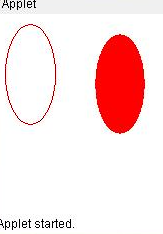
\*/

//draw filled oval

g.fillOval(100,20,50,100);

}

}



**Q39. Write Display Traffic Signal, that should be changing on every 1 min to red, yellow and green.**

public class LightThread {

public static void main(String[] args) {

Thread t1 = new Thread(new MyLight(),"t1");

t1.start();

}

}

public class MyLight extends Runnable {

public void run() {

while(true) {

System.out.println("Light: Yellow");

Thread.sleep(1000);

System.out.println("Light: Red");

Thread.sleep(1000);

System.out.println("Light: Green");

Thread.sleep(1000);

}

}

}

