ADVANCE JAVA ASSIGNMENT

NAME :- ARTI KUMARI

ROLL NO :-11920

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

import java.net.\*;

class adprgm1

{

public static void main(String args[])

throws UnknownHostException

{

InetAddress i = InetAddress.getByName("localhost");

System.out.println(i);

System.out.println("IP Address: "+i.getHostAddress()+"\nName: "+i.getHostName());

}

}

**OUTPUT**

C:\javaprograms\sem6>javac adprgm1.java

C:\javaprograms\sem6>java adprgm1

localhost/127.0.0.1

IP Address: 127.0.0.1

Name: localhost

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 q3

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

String str1=sc.next();

vowels v=new vowels(str1);

}

}

**OUTPUT**

C:\javaprograms\sem6>javac q3.java

C:\javaprograms\sem6>java q3

enter a string

ARTI

vowels are

a i

C:\javaprograms\sem6>

\*/

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

import java.applet.\*;

import java.awt.\*;

class adprgm5 extends Applet implements Runnable

{

Thread t;

int r,y,i,g1;

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

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

g.setColor(Color.yellow);

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

y = 0;

}

if(g1 == 1)

{

g.setColor(Color.black);

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

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

g.setColor(Color.green);

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

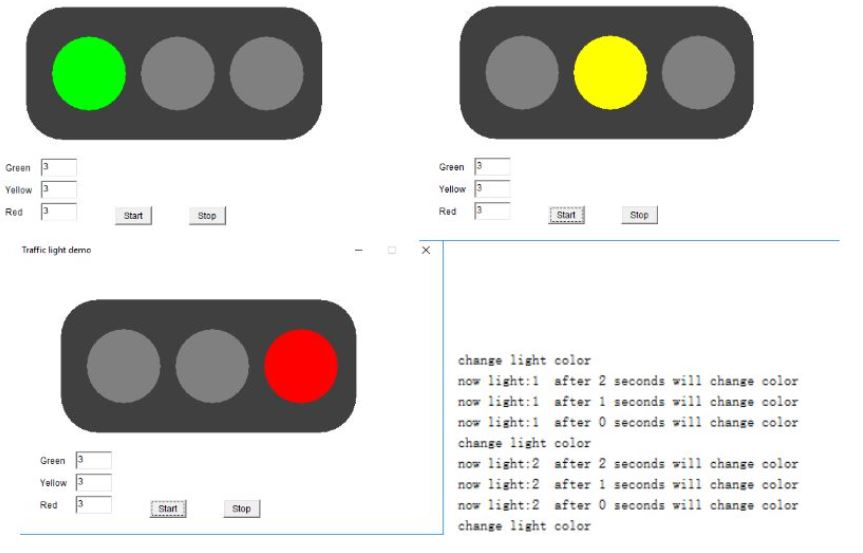
g1 = 0;

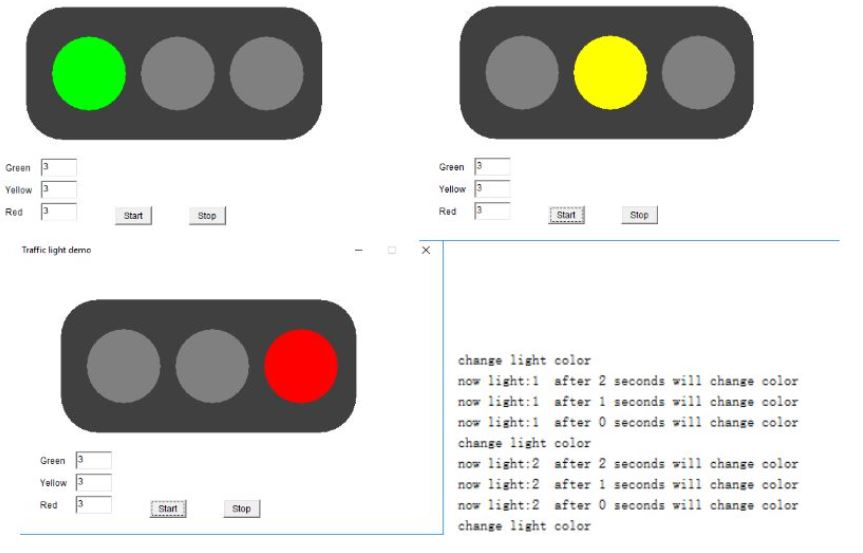
}

}

}

OUTPUT:





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

class Hellojava implements Runnable

{

Thread t;

public Hellojava(String title)

{

t=new Thread(this,title);

t.start();

}

public void run()

{

for(int i=0;i<20;i++)

{

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

try{Thread.sleep(100);}

catch(Exception e){}

}

}

}

public class adprgm6

{

public static void main(String a[])

{

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

Hellojava hj=new Hellojava("Hello java");

}

}

**OUTPUT**

C:\javaprograms\sem6>javac adprgm6.java

C:\javaprograms\sem6>java adprgm6

ThreadName: main

1ThreadName:Hello java

2ThreadName:Hello java

3ThreadName:Hello java

4ThreadName:Hello java

5ThreadName:Hello java

6ThreadName:Hello java

7ThreadName:Hello java

8ThreadName:Hello java

9ThreadName:Hello java

10ThreadName:Hello java

11ThreadName:Hello java

12ThreadName:Hello java

13ThreadName:Hello java

14ThreadName:Hello java

15ThreadName:Hello java

16ThreadName:Hello java

17ThreadName:Hello java

18ThreadName:Hello java

19ThreadName:Hello java

20ThreadName:Hello java

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

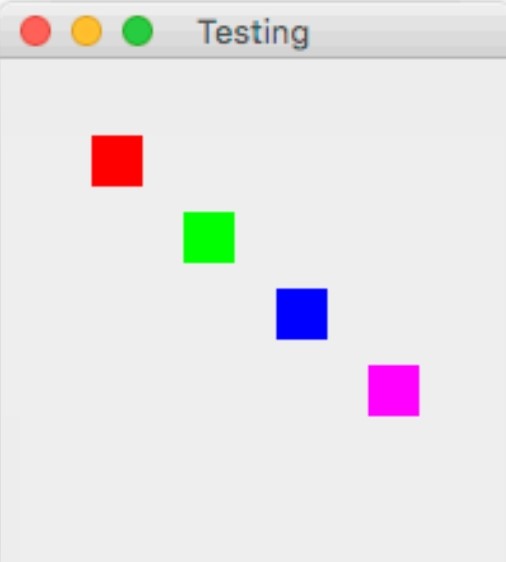
import java.awt.\*;  
import java.awt.event.\*;  
class MoveText extends Frame implements Runnable  
{  
 Label l1;  
 Thread t;  
 int x,y,side;  
 public MoveText()  
 {  
 setLayout(null);  
 l1=new Label(" Hello Java");  
 l1.setFont(new Font("",Font.BOLD,14));  
 l1.setForeground(Color.red);  
 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-=5,70,15);  
 add(l1);  
 if(y==20)  
 side=2;  
 }  
 if(side==2)  
 {  
 t.sleep(50);  
 l1.setBounds(x+=5,y+=5,70,15);  
 add(l1);  
 if(y==200)  
 side=3;  
 }  
 if(side==3)  
 {  
 t.sleep(50);  
 l1.setBounds(x-=5,y+=5,70,15);  
 add(l1);  
 if(y==390)  
 side=4;  
 }  
 if(side==4)  
 {  
 t.sleep(50);  
 l1.setBounds(x-=5,y-=5,70,15);  
 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 MoveText();  
 }  
}

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

import java.awt.\*;  
import java.applet.\*;  
/\*  
 <applet code= "CAR.java" height="600" width="600">  
 </applet>  
 \*/  
  
public class CAR extends java.applet.Applet implements Runnable {  
   
   
     int aflag,x;  
     Thread t;  
    public void init() {  
       
       t=new Thread(this); aflag=0;x=0;  
       t.start();  
       
    }  
   public void run()  
    {  
     try  
     {  
                x+=5;  
      t.sleep(200);  
        
        
        
      repaint();   
      run();  
     }  
     catch(Exception e)  
      {  
     }  
       
    }  
  
    public void paint(Graphics g) {  
       
     g.drawLine(0,150,600,150);  
     if (aflag==0)  
      { g.fillRect(x,100,20,20);  
             g.fillRect(x+10,125,20,20);  
             g.drawRect(x,100,20,20);  
             g.drawRect(x+10,125,20,20);  
      aflag=1;  
      }  
      else  
      {  
      g.fillRect(x,100,20,20);  
             g.fillRect(x,125,20,20);  
             g.drawRect(x,100,20,20);  
             g.drawRect(x,125,20,20);  
      aflag=0;  
      }

}  
}

OUTOUT



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

import java.awt.\*;

import java.awt.event.\*;

public class adprgm13 extends Frame implements Runnable

{

Thread t;

Label l1;

int f;

public adprgm13()

{

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();

}

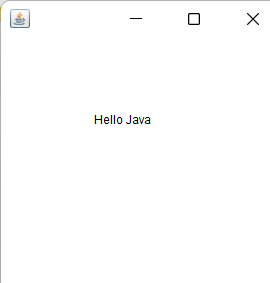
public static void main(String args[])

{

new adprgm13();

}

}



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

import java.awt.\*;

import java.awt.event.\*;

class adprgm15 extends Frame implements Runnable

{

Label l1;

Thread t;

intx,y,side;

adprgm15()

{

setLayout(null);

l1 = new Label(".");

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(y==0)

{

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

}

}

}

catch(Exception e)

{

System.out.println(e);

}

run();

}

public static void main(String args[])

{

new adprgm15();

}

}

}

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

import java.awt.\*;

/\*<applet code="BouncingBall.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++;     

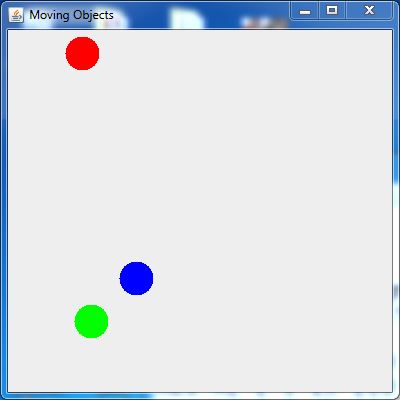
         f=0;

         }

        }

    }

}



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.\*;

public class adprgm18 extends Frame implements ActionListener, Runnable

{

Button b1,b2;

TextField t11,t12;

intcnt;

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

Thread t2 = new Thread(this,"t2");

public adprgm18()

{

setLayout(null);

t11 = new TextField();

t12 = new TextField();

b1 = new Button("Start");

b2 = new Button("Stop");

t11.setBounds(50,50,100,100);

t12.setBounds(160,50,100,100);

b1.setBounds(50,170,100,30);

b1.setBounds(160,170,100,30);

add(t11);

add(t12);

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(inti=1;i<=100;i++)

{

t11.setText("" + i);

t1.sleep(150);

t12.setText("" + i);

t2.sleep(150);

}

}

catch(Exception e)

{

}

}

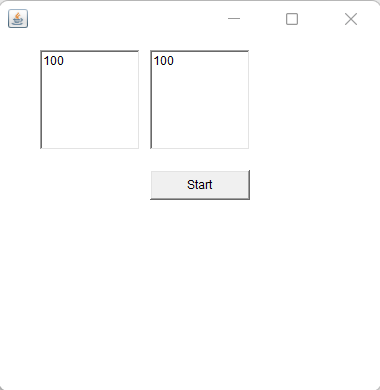
public static void main(String args[])

{

new adprgm18().show();

}

}



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 adprgm24

{

public static void main(String[] args)

{

MyThread t1 = new MyThread("ARTI") , t2 = new MyThread("POOJA");

t1.start();

t2.start();

try

{

t1.join();

t2.join();

}

catch(Exception e)

{

}

System.out.println(t1.getName() + "Thread dead");

System.out.println(t2.getName() + "Thread dead");

}

}

OUTPUT:

Thread[ARTI,5,main]

POOJA is sleeping for1700msec

ARTI is sleeping for1024msec

Thread[ARTI,5,main]

ARTI is sleeping for4377msec

Thread[POOJA,5,main]

POOJA is sleeping for4250msec

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);

}

}

}

**OUTPUT**

C:\javaprograms>javac MainThread.java

C:\javaprograms>java MainThread

Current Thread:Thread[main,5,main]

After the name is Changed:Thread[My Thread ,5,main]

2 1

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(inti=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);

}

}

OUTPUT:

FIRST1

FIRST2

FIRST3

FIRST4

SECOND1

SECOND2

SECOND3

SECOND4

SECOND5

THIRD1

THIRD2

FIRST5

THIRD3

THIRD4

THIRD5

Q29. Write a Multithreading program in java to convert smile face into the crying face after 5 seconds and vice versa(Use Applet).

/\*<applet code= "adprgm29.class" height="300" width="300"></applet>\*/import java.awt.\*;

import java.applet.\*;

public class adprgm29 extends Applet implements Runnable

{

intaflag;

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);

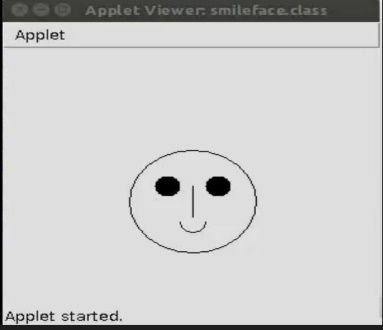
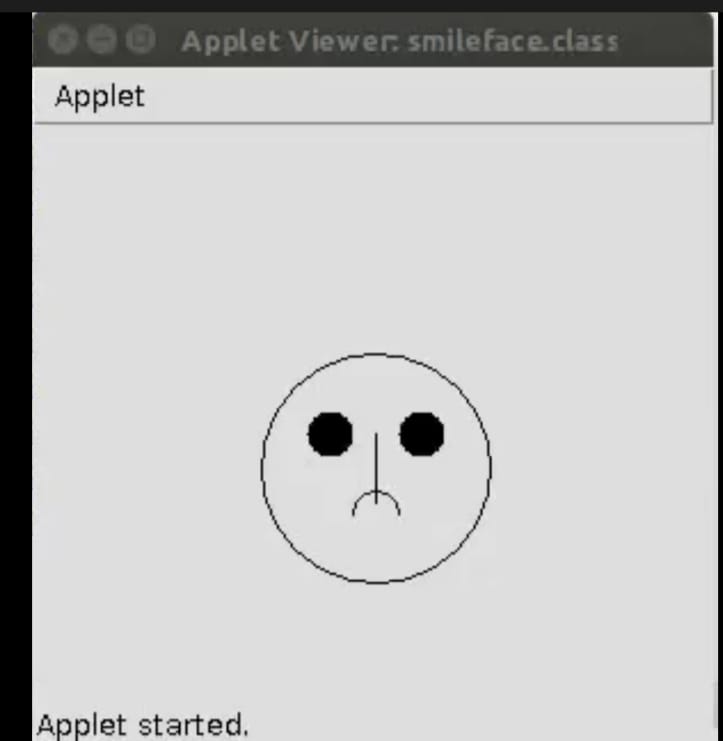
aflag=0;

}

}

}

OUTPUT

Q-37. Write a applate a programe to create a oval animation starting from samll oval and after every time instance one size bigger oval should be drawn making a shape of cone.

<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){

setForeground(Color.red);

To draw a oval in an applet window use

drawOval(int x1,int y1, int width, int height)

Method.

This method draws a oval of specified width and

Height at (x1,y1)

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

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

}

}

Q-39. Wrtite 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(2000);

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

Thread.sleep(2000);

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

Thread.sleep(2000);

}

}

}

Q-40. Write a Multithreading program in java to display all the alphabets from A to Z after 3 seconds.

public class adprgm40 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[])

{

adprgm40 t=new adprgm40();

t.start();

}

}

/\*

C:\javaprograms\sem6>java adprgm40

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Q-43. 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 adprgm43 extends Frame implements ActionListener, Runnable

{

Button start,stop;

TextFieldtf;

int x = 0,y = 0;

String msg = "";

Thread t1 = new Thread(this);

public adprgm43()

{

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)

{

intsec,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);

}

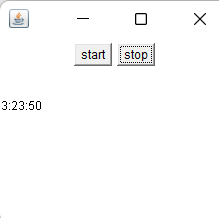
public static void main(String args[])

{

new adprgm43();

}

}



Q-45. 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.\*;

import java.applet.\*;

public class adprgm45 extends Applet

{

static int N = 10;

public void paint(Graphics g)

{

intx,y;

for(int row = 0; row < N; row++)

{

for(int col = 0; col < N; col++)

{

x = row \* 20;

y = col \* 20;

if((row % 2 == 0) == (col % 2 == 0))

g.setColor(Color.BLACK);

else

g.setColor(Color.WHITE);

g.fillRect(x, y, 20, 20);

}

}

}

}

