9) a) **Aim:** Develop a Java application for the blinking eyes and mouth should open while blinking.

Description:

→ we can use Applet concepts for blinking eyes

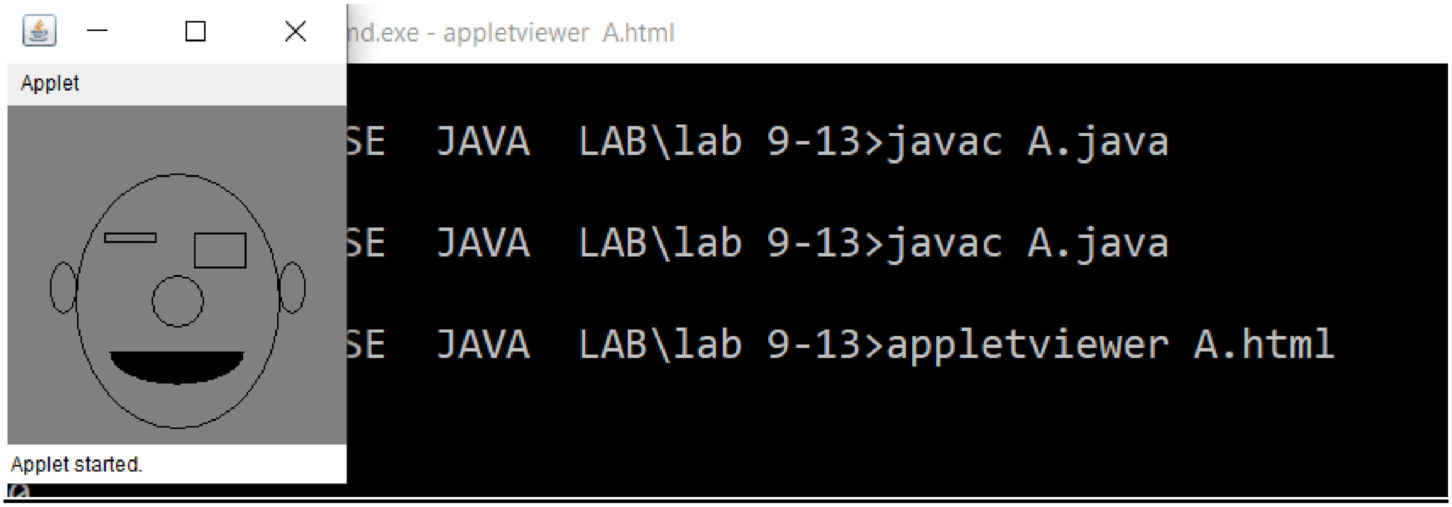
→ we can use paint() method and drawstring() methods

→ we draw symbols like rectangle, oval like many for many operations in this code

Program:

```
import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Canvas;
import java.awt.Color;
import java.awt.Graphics;

public class A extends Applet
{
    private static final long serialVersionUID = -1152278362796573663L;

    public class MyCanvas extends Canvas
   {
      private static final long serialVersionUID = -4372759074220420333L;
      private int flag = 0;
      public void paint(Graphics g)
     {
         draw();
     }

     public void draw()
    {
         Graphics g = this.getGraphics();
         g.setColor(Color.BLACK);
         super.paint(g);
         if (flag == 0)
         {
                 System.out.println(flag);
                 g.drawOval(40, 40, 120, 150);// face
                 g.drawRect(57, 75, 30, 5);// left eye shut
                 g.drawRect(110, 75, 30, 20);// right eye
                 g.drawOval(85, 100, 30, 30);// nose
                 g.fillArc(60, 125, 80, 40, 180, 180);// mouth
                 g.drawOval(25, 92, 15, 30);// left ear
                 g.drawOval(160, 92, 15, 30);// right ear
                 flag = 1;
         }
         else
         {
                 System.out.println(flag);
                 g.drawOval(40, 40, 120, 150);// face
                 g.drawOval(57, 75, 30, 20);// left eye
                 g.drawOval(110, 75, 30, 20);// right eye
                 g.fillOval(68, 81, 10, 10);// left pupil
                 g.fillOval(121, 81, 10, 10);// right pupil
                 g.drawOval(85, 100, 30, 30);// nose
                 g.fillArc(60, 125, 80, 40, 180, 180);// mouth
                 g.drawOval(25, 92, 15, 30);// left ear
                 g.drawOval(160, 92, 15, 30);// right ear
                 flag = 0;
         }

         try
         {
                 Thread.sleep(900);
         }
         catch (Exception e)
         {
                 System.out.println("killed while sleeping");
         }
                 this.repaint(100);
   }
}
         public void init()
         {
                 this.C = new MyCanvas();
                 this.setLayout(new BorderLayout());
                 this.add(C, BorderLayout.CENTER);
                 C.setBackground(Color.GRAY);
         }
         private MyCanvas C;
}
/*
<applet     code  =  "A.class"    width=200   height=200>
</applet>
               */
```

Output:



```
SE  JAVA  LAB\lab 9-13>javac A.java

SE  JAVA  LAB\lab 9-13>javac A.java

SE  JAVA  LAB\lab 9-13>appletviewer A.html
```

Applet started.

9 b) Aim: Develop a Java application that simulates a traffic light.

The program lets the user select one ofthe three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with ―STOP‖ or ―READY‖ or ‖GO‖ should appear above the buttons in the selected color.Initially, there is no message shown

Description:

We can use different classes related to Applets and its methods

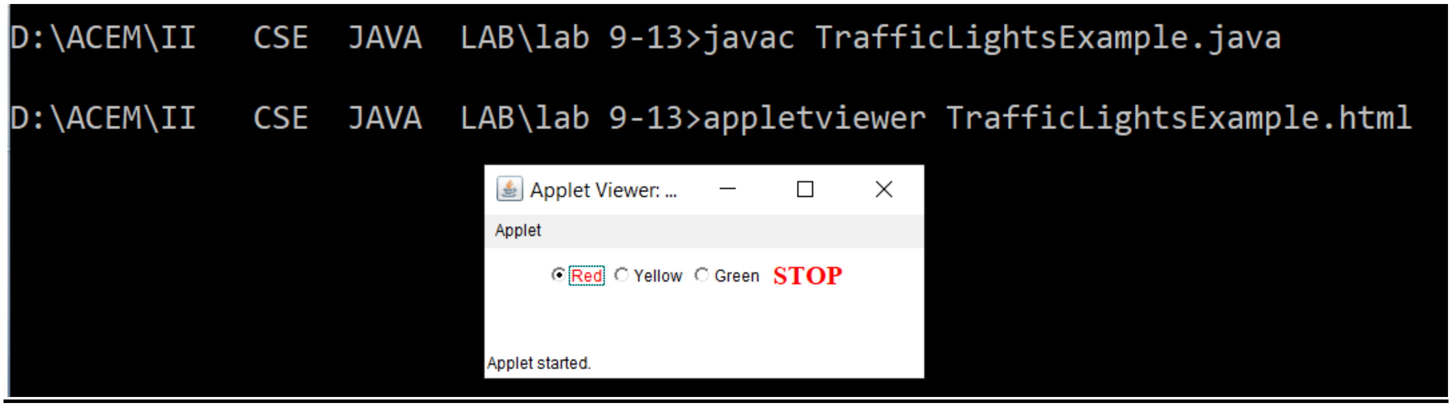
Write init() method to write code

Program:

```
import java.applet.Applet;

import java.awt.*;

import java.awt.event.*;


public class TrafficLightsExample extends Applet implements ItemListener
{
        CheckboxGroup grp = new CheckboxGroup();

        Checkbox redLight, yellowLight, greenLight;

        Label msg;

        public void init()
        {
                redLight = new Checkbox("Red", grp, false);

                yellowLight = new Checkbox("Yellow", grp, false);

                greenLight = new Checkbox("Green", grp, false);


                msg = new Label("");
                redLight.addItemListener(this);
                yellowLight.addItemListener(this);
                greenLight.addItemListener(this);

                add(redLight);
                add(yellowLight);
                add(greenLight);

                add(msg);
                msg.setFont(new Font("Serif", Font.BOLD, 20));
        }

        public void itemStateChanged(ItemEvent ie)
        {
                redLight.setForeground(Color.BLACK);
                yellowLight.setForeground(Color.BLACK);
                greenLight.setForeground(Color.BLACK);

                if(redLight.getState() == true)
                {
                    redLight.setForeground(Color.RED);
                    msg.setForeground(Color.RED);
                    msg.setText("STOP");
                }
                else
                if(yellowLight.getState() == true)
                {
                    yellowLight.setForeground(Color.YELLOW);
                    msg.setForeground(Color.YELLOW);
                    msg.setText("READY");
                }
                else
                {
                    greenLight.setForeground(Color.GREEN);
                    msg.setForeground(Color.GREEN);
                    msg.setText("GO");
                }
        }
}
```

```
/*
 <applet code = "TrafficLightsExample.class"  width = 1000  height = 500>
 </applet>     */
```

Output:



```
D:\ACEM\II   CSE   JAVA  LAB\lab 9-13>javac TrafficLightsExample.java

D:\ACEM\II   CSE   JAVA  LAB\lab 9-13>appletviewer TrafficLightsExample.html
```

Applet started.

10 a) Aim: Develop a Java application to implement the opening of a door while opening man should present before hut and closing man should disappear

Description:

- Use awt for applet classes
- Use animation classes
- ImageIcon for image in the below code

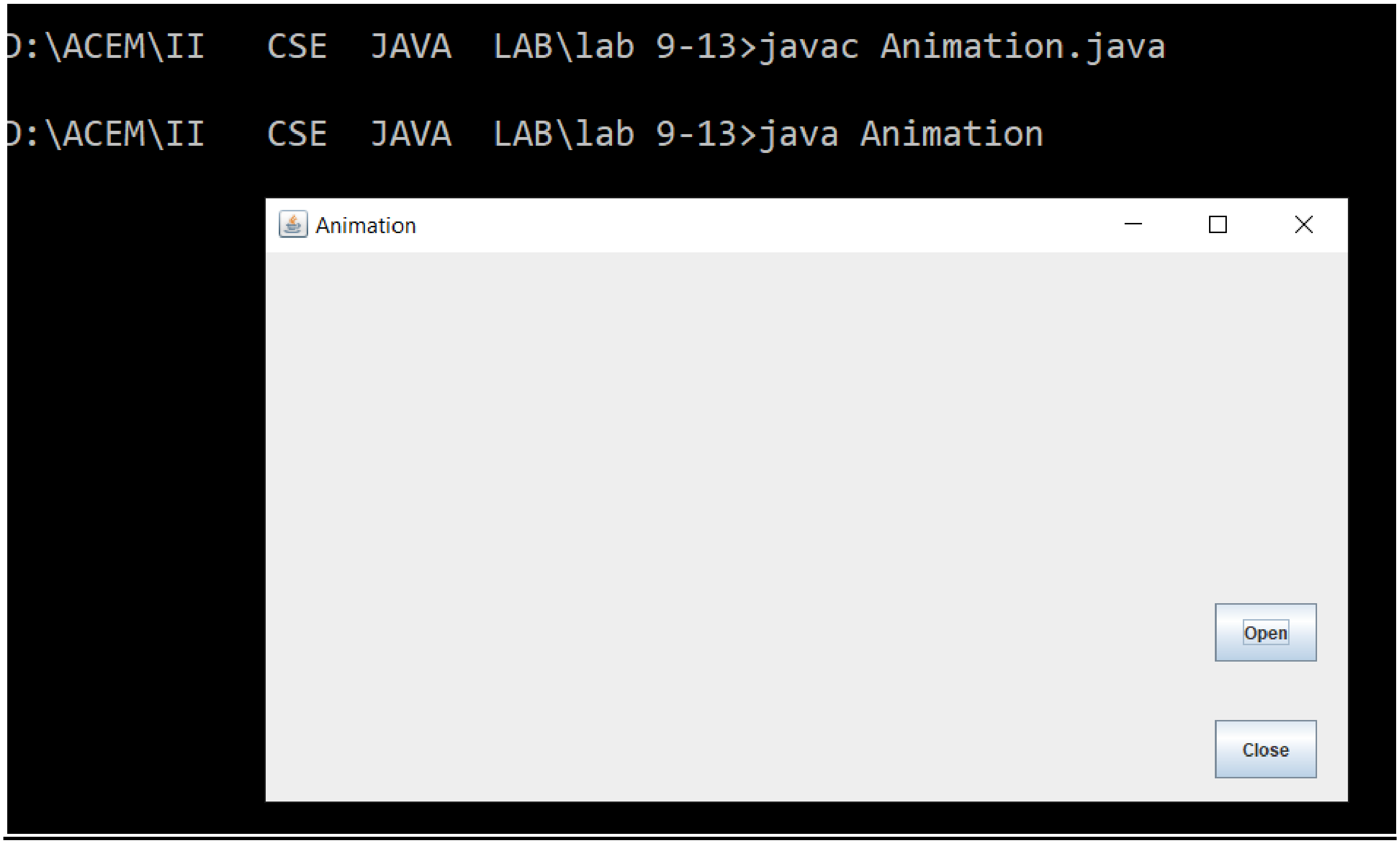
Program:

```
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
class Animation extends JFrame implements ActionListener
{

        ImageIcon ii1, ii2;

        Container c;

        JButton b1,b2;

        JLabel lb1;
```

```
Animation()
{
        c = getContentPane();

        c.setLayout(null);

        ii1 = new ImageIcon("house0.jpg");

        ii2 = new ImageIcon("house1.jpg");

        lb1 = new JLabel(ii1);

        lb1.setBounds(50,10,500,500);



        b1 = new JButton("Open");

        b2 = new JButton("Close");



        b1.addActionListener(this);

        b2.addActionListener(this);



        b1.setBounds(650,240,70,40);

        b2.setBounds(650,320,70,40);



        c.add(lb1);

        c.add(b1);

        c.add(b2);
}

public void actionPerformed(ActionEvent ae)
{
        String str = ae.getActionCommand();
        if( str.equals("Open") )

        lb1.setIcon(ii2);

        else

        lb1.setIcon(ii1);

}
```

```
    public static void main(String args[])
    {
        Animation ob = new Animation();
        ob.setTitle("Animation");

        ob.setSize(800,600);

        ob.setVisible(true);

        ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

}
```

Output:



```
D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>javac Animation.java

D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>java Animation
```

10 b) **Aim:** Develop a Java application by using JtextField to read decimal values and converting a decimal number into a binary number then print the binary value in another JtextField.

Description:

Binary to decimal conversion is done to convert a number given in the binary system to its equivalent in the decimal number system.

A number system is a format to represent numbers in a certain way.

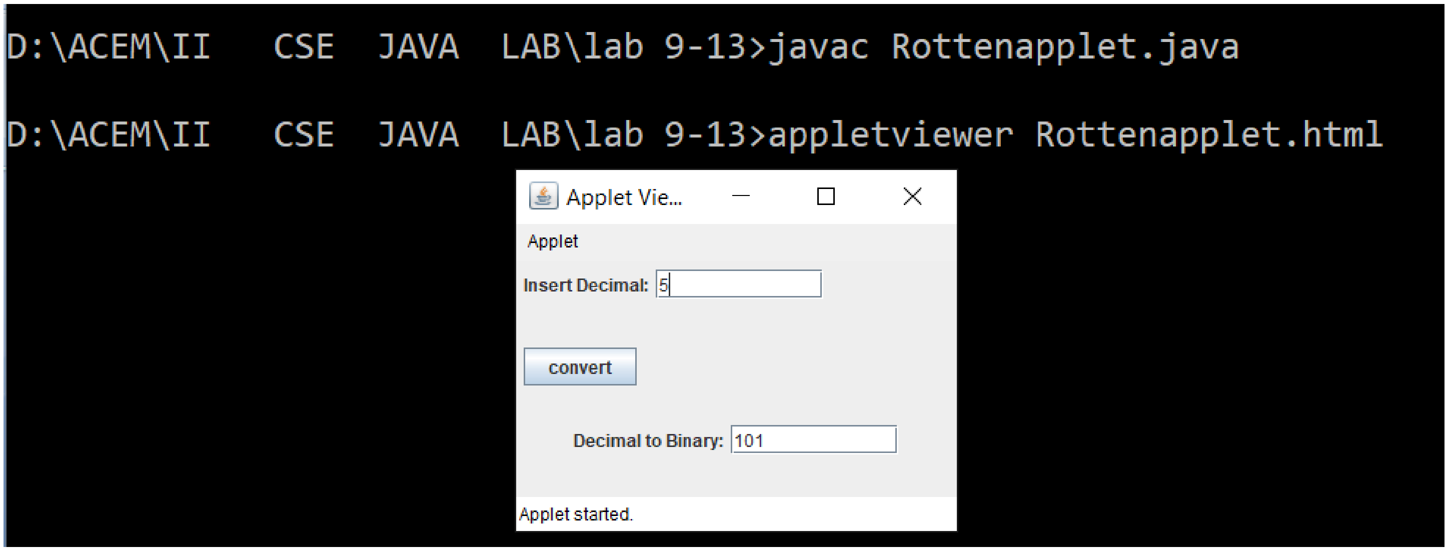
Binary Number System – The binary number system is used in computers and electronic systems to represent data, and it consists of only two digits which are 0 and 1.

Input: 33 **Output**: 100001

Program:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
public class Rottenapplet extends JApplet implements ActionListener
{
        JPanel mainpanel=new JPanel(new GridLayout (3,1));

        JPanel p1=new JPanel(new FlowLayout(0));
        JPanel p2=new JPanel(new FlowLayout (0));
        JPanel p3=new JPanel(new FlowLayout ());


        JTextField q1=new JTextField (10);
        JTextField q2=new JTextField (10);

        JButton clickbutton = new JButton("convert");

        public void init()
        {
                getContentPane().add(mainpanel);
                mainpanel.add(p1);
                mainpanel.add(p2);
                mainpanel.add(p3);


                p1.add(new JLabel("Insert Decimal:"));


                p1.add(q1);
                p2.add(clickbutton);
                p3.add(new JLabel("Decimal to Binary:"));
                p3.add(q2);


                clickbutton.addActionListener(this);
        }


        public void actionPerformed(ActionEvent x)
        {
                if(x.getSource()==clickbutton)
                {
                        int counter,dec,user;
                        user=Integer.valueOf(q1.getText()).intValue();


                        String[]conversion=new String[8];
                        String[]complete=new String[4];
                        counter=0;
                        complete[0]="";


                        do
                        {
                           dec=user%2;
                           conversion[counter]=String.valueOf(dec);
                           complete[0]=conversion[counter]+complete[0];
                           user=user/2;
                           counter=counter+1;
                        }


                   while(user !=0);
                   q2.setText(String.valueOf(complete[user]));
                }
          }
}
/*
       <applet    code="Rottenapplet.class"    height=300    width=300>
      </applet>    */
```

Output:



```
D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>javac Rottenapplet.java

D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>appletviewer Rottenapplet.html
```

Insert Decimal: 5

Decimal to Binary: 101

Applet started.

11 a) **Aim:** Develop a Java application that handles all mouse events and shows the event name at the centerof the window when a mouse event is fired. Use adapter classes

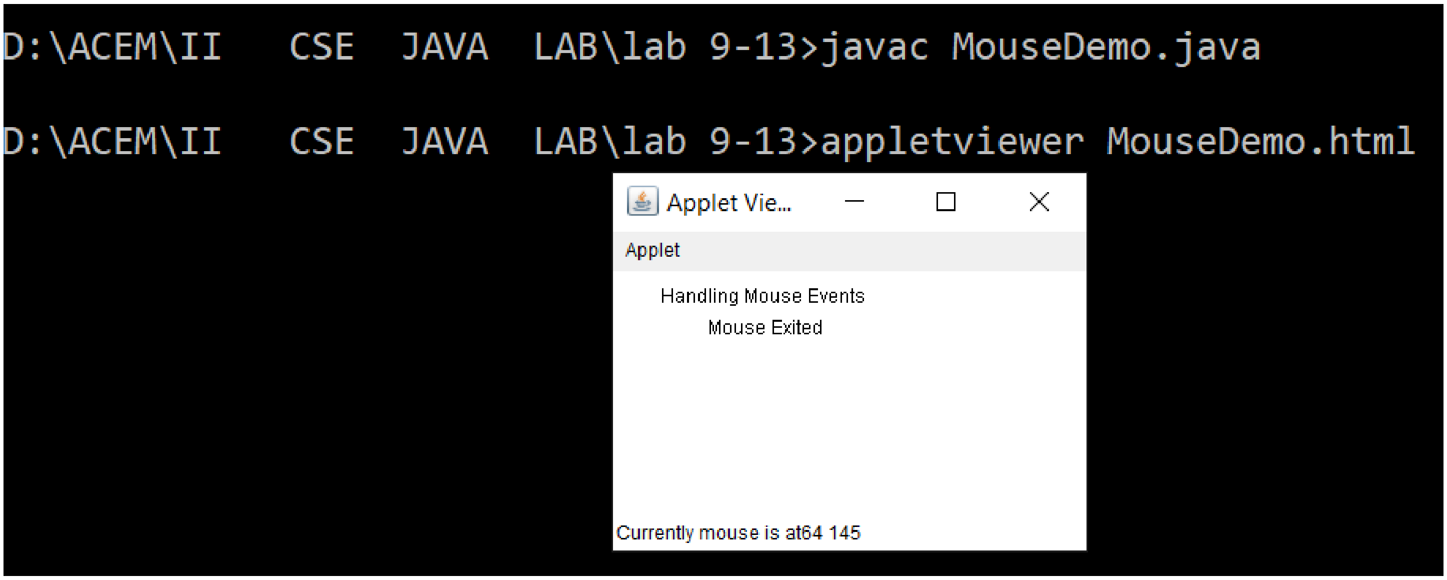
Description:

- click: click event occurs when mouse is clicked on the register element. ...
- mouseup: mouseup event occurs when button of the mouse is released over an element. ...
- mousedown: mousedown event occurs when button of the mouse is pressed over an element. ...
- mousemove: mousemove event occurs when button of the mouse move over an element. ...
- mouseover: mouseover event occurs when the mouse cursor moves onto the element. ...

Program:

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

public class MouseDemo extends Applet implements MouseListener, MouseMotionListener
{
        int mx=0;
        int my=0;

        String msg="";

        public void init()
        {
                addMouseListener(this);
                addMouseMotionListener(this);
        }
        public void mouseClicked(MouseEvent me)
        {
                mx=20;
                my=40;
                msg="Mouse Clicked";
                repaint();
        }

        public void mousePressed(MouseEvent me)
        {
                mx=30;
                 my=60;
                msg="Mouse Pressed";
                repaint();
        }

        public void mouseReleased(MouseEvent me)
        {
                mx=30;
                my=60;
                msg="Mouse Released";
                repaint();
        }

        public void mouseEntered(MouseEvent me)
        {
                mx=40;
                 my=80;
                msg="Mouse Entered";
                repaint();
        }

        public void mouseExited(MouseEvent me)
        {
                mx=40;
                my=80;
                msg="Mouse Exited";
                repaint();
        }

        public void mouseDragged(MouseEvent me)
        {
                mx=me.getX();
                my=me.getY();
                showStatus("Currently mouse dragged"+mx+" "+my);
                repaint();
        }

        public void mouseMoved(MouseEvent me)
        {
                mx=me.getX();
                my=me.getY();
                showStatus("Currently mouse is at"+mx+" "+my);
                repaint();
        }

        public void paint(Graphics g)
        {
                g.drawString("Handling Mouse Events",30,20);
                g.drawString(msg,60,40);
        }
}
/*
    <applet    code="MouseDemo.class" width=300 height=300>
    </applet>    */
```

Output:



```
D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>javac MouseDemo.java

D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>appletviewer MouseDemo.html
```

```
Handling Mouse Events
      Mouse Exited

Currently mouse is at64 145
```

11 b) **Aim:** Develop a Java application to demonstrate the key event handlers

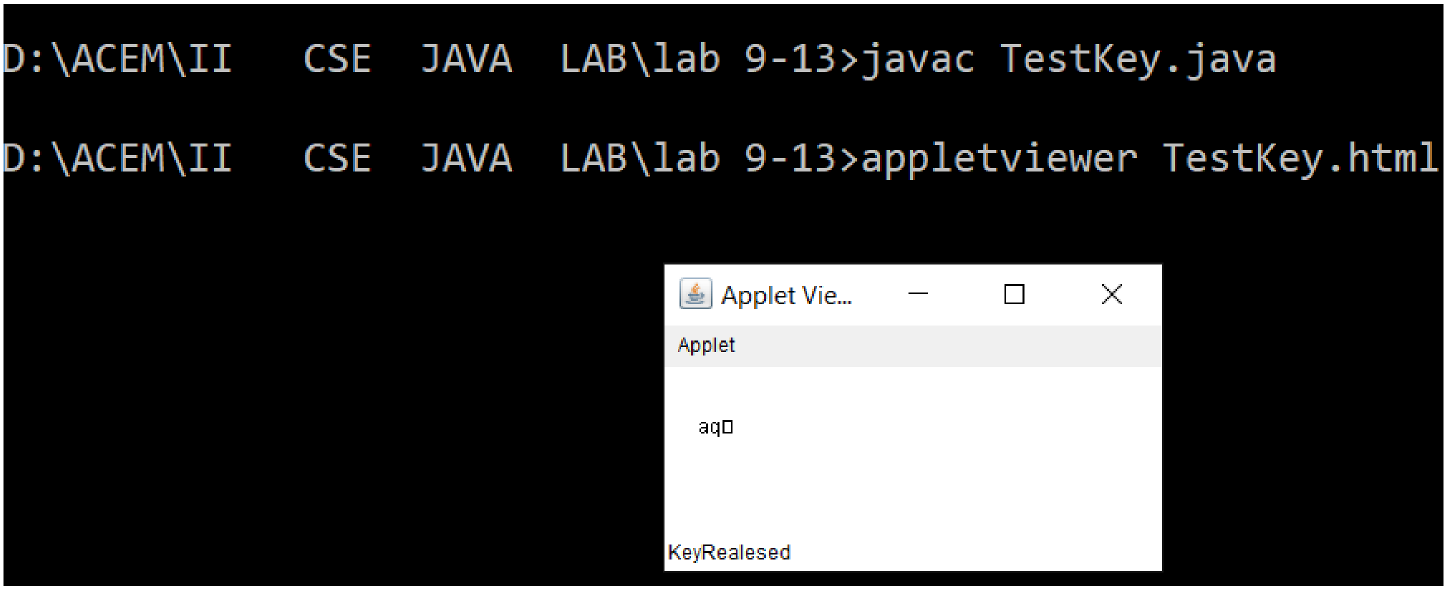
Description:

An **event** can be defined as changing the state of an object or behavior by performing actions. Actions can be a button click, cursor movement, keypress through keyboard or page scrolling, etc.

Program:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.applet.*;
import java.awt.event.*;
import java.awt.*;
public class TestKey extends Applet implements KeyListener
{
        String msg="";

        public void init()
        {
                addKeyListener(this);
        }

        public void keyPressed(KeyEvent k)
        {
                showStatus("KeyPressed");
        }

        public void keyReleased(KeyEvent k)
        {
                showStatus("KeyRealesed");
        }

        public void keyTyped(KeyEvent k)
        {
                msg = msg+k.getKeyChar();
                repaint();
        }

        public void paint(Graphics g)
        {
                g.drawString(msg, 20, 40);
        }
}
/*
<applet code="TestKey.class" width=300   height=100>
</applet>   */
```

Output:



```
D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>javac TestKey.java

D:\ACEM\II   CSE   JAVA   LAB\lab 9-13>appletviewer TestKey.html
```

```
aq□

KeyRealesed
```

12 A) **Aim**: Develop a Java application to find the maximum value from the given type of elements using a generic function

Description:

1. Create a class Myclass to implement generic class and generic methods.
2. Get the set of the values belonging to specific data type.
3. Create the objects of the class to hold integer,character and double values.
4. Create the method to compare the values and find the maximum value stored in the array.
5. Invoke the method with integer, character or double values . The output will be displayed based on the data type passed to the method.

Program:

```
class   MyClass<T extends Comparable<T>>
{
        T[] vals;
```

```
    MyClass(T[]  o)
    {
        vals = o;
    }

    public T min()
    {
        T  v = vals[0];

        for(int i=1; i < vals.length; i++)
        if(vals[i].compareTo(v) < 0)
        v = vals[i];
        return v;
    }

    public T max()
    {
        T v = vals[0];
        for(int i=1; i < vals.length;i++)
        if(vals[i].compareTo(v) > 0)
        v = vals[i];
        return v;
    }
}

class MaxMinUseGeneric
{
    public static void main(String args[])
    {
        int i;

        Integer   inums[]  =  {10,2,5,4,6,1};
        Character chs[]={'v','p','s','a','n','h'};
        Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};

        MyClass<Integer> iob = new MyClass<Integer>(inums);
        MyClass<Character> cob = new MyClass<Character>(chs);
        MyClass<Double>dob = new MyClass<Double>(d);

        System.out.println("Max value in inums: " + iob.max());
        System.out.println("Min value in inums: " + iob.min());
        System.out.println("Max value in chs: " + cob.max());
        System.out.println("Min value in chs: " + cob.min());
        System.out.println("Max value in chs: " + dob.max());
```

```
                    System.out.println("Min value in chs: " + dob.min());
          }
}
```

Output:

```
D:\ACEM\II   CSE  JAVA  LAB\lab 9-13>javac MaxMinUseGeneric.java
D:\ACEM\II   CSE  JAVA  LAB\lab 9-13>java MaxMinUseGeneric

Max value in inums: 10
Min value in inums: 1
Max value in chs: v
Min value in chs: a
Max value in chs: 88.3
Min value in chs: 10.4
```

12 b) **Aim:** Develop a Java application that works as a simple calculator. Use a grid layout to arrange buttonsfor the digits and for the +, -,*, % operations. Add a text field to display the result

Description:

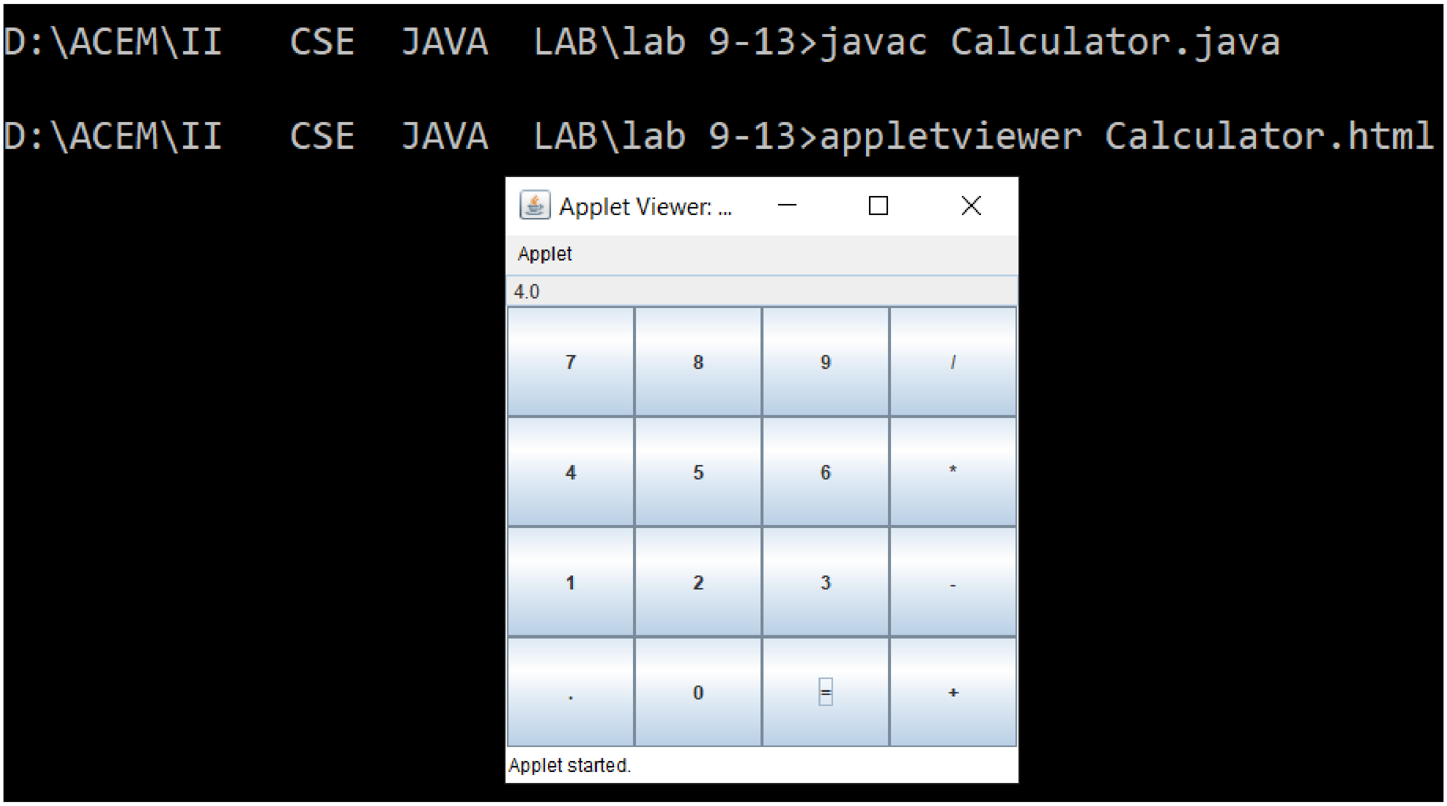
GridLayout is one of the Layout managers.
A layout manager automatically arranges your controls with in a window by using some type of algorithm.
Grid Layout lays out component in a two dimensional grid.
When you instantiate a GridLayout,you define the number of rows and columns

Program:

```
import javax.swing.*;

import java.awt.*;

import

java.awt.event.*;

 public class Calculator extends JApplet
{
          public void init()
          {
                    CalculatorPanel calc=new CalculatorPanel();
                    getContentPane().add(calc);
          }
}

class CalculatorPanel extends JPanel implements ActionListener
{

   JButton        n1,n2,n3,n4,n5,n6,n7,n8,n9,n0,plus,minus,mul,div,dot,equal;

   static JTextField result=new JTextField("0",45);
   static String lastCommand=null;
   JOptionPane p=new JOptionPane();
   double preRes=0,secVal=0,res;
   private  static void assign(String no)
  {

          if((result.getText()).equals("0"))

          result.setText(no);

          else if(lastCommand=="=")
          {
                    result.setText(no);
                    lastCommand=null;
          }
          else
                    result.setText(result.getText()+no);
   }

     public CalculatorPanel()
    {
          setLayout(new BorderLayout());

          result.setEditable(false);

          result.setSize(300,200);

          add(result,BorderLayout.NORTH);


          JPanel panel=new JPanel();
          panel.setLayout(new   GridLayout(4,4));

          n7=new JButton("7");
          panel.add(n7);
          n7.addActionListener(this);


          n8=new JButton("8");
          panel.add(n8);
          n8.addActionListener(this);

          n9=new JButton("9");
          panel.add(n9);
          n9.addActionListener(this);

          div=new JButton("/");
          panel.add(div);
          div.addActionListener(this);

          n4=new JButton("4");
          panel.add(n4);
          n4.addActionListener(this);
          n5=new JButton("5");
          panel.add(n5);
          n5.addActionListener(this);

          n6=new JButton("6");
          panel.add(n6);
          n6.addActionListener(this);

          mul=new JButton("*");
          panel.add(mul);
          mul.addActionListener(this);

          n1=new JButton("1");
          panel.add(n1);
          n1.addActionListener(this);

          n2=new JButton("2");
          panel.add(n2);
          n2.addActionListener(this);

          n3=new JButton("3");
          panel.add(n3);
          n3.addActionListener(this);

          minus=new JButton("-");
          panel.add(minus);
          minus.addActionListener(this);

          dot=new JButton(".");
          panel.add(dot);
          dot.addActionListener(this);

          n0=new JButton("0");
          panel.add(n0);
          n0.addActionListener(this);

          equal=new JButton("=");
          panel.add(equal);
          equal.addActionListener(this);

          plus=new JButton("+");
          panel.add(plus);
          plus.addActionListener(this);

          add(panel,BorderLayout.CENTER);

  }
```

```
public void actionPerformed(ActionEvent ae)
{
        if(ae.getSource()==n1)
                assign("1");

        else   if(ae.getSource()==n2)

                assign("2");

        else    if(ae.getSource()==n3)

                assign("3");

        else     if(ae.getSource()==n4)

                assign("4");

        else     if(ae.getSource()==n5)

                assign("5");

        else      if(ae.getSource()==n6)

                assign("6");

        else      if(ae.getSource()==n7)

                        assign("7");

        else      if(ae.getSource()==n8)

                assign("8");

        else      if(ae.getSource()==n9)

                assign("9");

        else      if(ae.getSource()==n0)

                assign("0");

        else      if(ae.getSource()==dot)
        {
                if(((result.getText()).indexOf("."))==-1)
                result.setText(result.getText()+".");

        }

        else       if(ae.getSource()==minus)
        {
                preRes=Double.parseDouble(result.getText());
                 lastCommand="-";

                result.setText("0");

        }
        else if(ae.getSource()==div)
          {
                  preRes=Double.parseDouble(result.getText());
                  lastCommand="/";

                  result.setText("0");

          }

          else if(ae.getSource()==equal)
          {
                  secVal=Double.parseDouble(result.getText());
                  if(lastCommand.equals("/"))

                          res=preRes/secVal;

                  else   if(lastCommand.equals("*"))

                                  res=preRes*secVal;

                  else   if(lastCommand.equals("-"))

                          res=preRes-secVal;

                  else   if(lastCommand.equals("+"))

                          res=preRes+secVal;

                          result.setText(" "+res);

                          lastCommand="=";
          }
          else
          if(ae.getSource()==mul)
          {
                  preRes=Double.parseDouble(result.getText());
                  lastCommand="*";

                  result.setText("0");
          }
          else
          if(ae.getSource()==plus)
          {
                  preRes=Double.parseDouble(result.getText());
                  lastCommand="+";
                  result.setText("0");
          }
      }
}

/*
      <applet       code="Calculator.class"  height=300 width=200>
      </applet>
                                  */
```

Output:



```
D:\ACEM\II   CSE  JAVA  LAB\lab 9-13>javac Calculator.java

D:\ACEM\II   CSE  JAVA  LAB\lab 9-13>appletviewer Calculator.html
```

```
4.0
Applet started.
```

13 **Aim**: . Develop a Java application to establish a JDBC connection, create a table student with propertiesname, register number, mark1, mark2, mark3. Insert the values into the table by using java and display the information of the students at front end.

Description:

Steps For Connectivity Between Java Program and Database

1. Import the Packages
2. Load the drivers using the *forName() method*
3. Register the drivers *using DriverManager*
4. Establish a connection *using the Connection class object*
5. Create a statement
6. Execute the query
7. Close the connections

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class TestApplication99
 {
     static final String  url = "jdbc:oracle:thin:@localhost:1521:XE";
```

```
   static final String user = "system";
   static final String pwd = "acem";

 public static void main(String[] args)
 {
        try
       {
        Connection   conn   =   DriverManager.getConnection(url,user,pwd);
        Statement    stmt    =    conn.createStatement();

                     String    sql    =    "CREATE    TABLE     SemOne(" +
                                         "name  char(15) , "+
                                                   "htno     int ,"+
                                         " marks1   int ,"+
                                                        " marks2    int ,"+
                                         " marks3    int)";

                      stmt.executeUpdate(sql);

                      System.out.println("Created table in given database...");
        //insertion o record
         PreparedStatement   ps  =  conn.prepareStatement("insert into SemOne  values(?,?,?,?,?)");
                         ps.setString(1, "charan");//1 specifies the first parameter in the query
                          ps.setInt(2,2001);
                         ps.setInt(3,88);
                         ps.setInt(4,99);
                         ps.setInt(5,95);
                        int   i  =  ps.executeUpdate();
                       System.out.println(i+" records inserted");

        //selecting records from database
        PreparedStatement    ps1    =    null;
                  String  qry    =   "select * from SemOne";

                  ps1     =    conn.prepareStatement(qry);

                  ResultSet    rs     =  ps1.executeQuery(qry);

                            while (rs.next())
                            {
                              String   name     =    rs.getString("name");
                              int    htno          =    rs.getInt("htno");
                              int     marks1    =    rs.getInt("marks1");
                              int     marks2    =    rs.getInt("marks2");
                              int     marks3    =    rs.getInt("marks3");
```

```
                                System.out.println("stu name : " + name);
                                System.out.println("studtno : " + htno);
                                System.out.println("sub1 : " + marks1);
                                System.out.println("sub2 : " + marks2);
                                System.out.println("sub3 : " + marks3);
                              }

        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
    }
}
```

<u>Output:</u>

```
D:\ACEM\II   CSE  JAVA  LAB\lab 9-13>javac TestApplication99.java
D:\ACEM\II   CSE  JAVA  LAB\lab 9-13>java TestApplication99

Created table in given database...

1 records inserted

stu name : charan
studtno : 2001
sub1 : 88
sub2 : 99
sub3 : 95
```