**3a)** Java applet program that displays a simple message

**import** java.awt.\*;

**import** java.applet.\*;

*/\**

*<applet code="sim" width=300 height=300>*

*</applet>*

*\*/*

**public** **class** sim **extends** Applet

{

String msg=" ";

**public** **void** init()

{

msg+="init()--->";

setBackground(Color.orange);

}

**public** **void** start()

{

msg+="start()--->";

setForeground(Color.blue);

}

**public** **void** paint(Graphics g)

{

msg+="paint()--->";

g.drawString(msg,200,50);

}

}

**Java applet program that displays a simple message**

**3b)**

**/\*<applet code=”myevent11″ width =1000 height =1000></applet>\*/**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.applet.\*;**

**public class myevent11 extends Applet implements ActionListener**

**{** **Label   l1,l2;**

**Button b1;**

**TextField t1,t2;**

**public void init()**

**{**

**setLayout(null);**

**l1 = new Label(“Enter the number”);**

**l2 = new Label(“Result”);**

**t1 = new TextField(30);**

**t2 = new TextField(30);**

**b1 = new Button(“Compute”);**

**add(t1);**

**add(t2);**

**add(l1);**

**add(l2);**

**add(b1);**

**l1.setBounds(50,100,150,30);**

**l2.setBounds(50,150,150,30);**

**t1.setBounds(250,100,150,30);**

**t2.setBounds(250,150,150,30);**

**b1.setBounds(230,240,100,30);**

**b1.addActionListener(this);**

**}**

**public void actionPerformed(ActionEvent e)**

**{**

**if(e.getActionCommand()==”Clicked”)**

**{**

**int a = Integer.parseInt(t1.getText());**

**int fact =1;**

**for (int i= 1; i<=a; i++)**

**{**

**fact=fact\*i;**

**}**

**t2.setText(String.valueOf(fact));**

**}**

**}**

**}**

**4)** Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

import java.awt.\*;

import javax.swing.\*;

import java.applet.Applet;

import java.awt.event.\*;

public class Division extends Applet implements ActionListener{

TextField t1,t2,t3;

Button b;

Label L1,L2,L3,L4;

String s;

Division e;

public void init()

{

e=this;

t1=new TextField(10);

t2=new TextField(10);

t3=new TextField(10);

L1=new Label("enter num1");

L2=new Label("enter num2");

L3=new Label("Result is");

L4=new Label("Division of 2 numbers");

b=new Button("Divide");

add(L4);

add(L1);

add(t1);

add(L2);

add(t2);

add(L3);

add(t3);

add(b);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

try

{

int num1=Integer.parseInt(t1.getText());

int num2=Integer.parseInt(t2.getText());

s=""+(num1/num2);

t3.setText(s);

}

catch(ArithmeticException a)

{

JOptionPane.showMessageDialog(null,"Divide by zero");

}

catch(NumberFormatException b)

{

JOptionPane.showMessageDialog(null,"NumberFormateException");

}

}

}

# 5) Write a java program that implements a simple/client application. That client sends data to a server. The server receives the data, Uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle(Use java.net)

//Program for implementing Server/Client Application

// Server Program

import java.io.\*;

import java.net.\*;

class Server

{

            public static void main(String args[])

            {

                        try

                        {

                                    ServerSocket ss=new ServerSocket(1064);

                                    System.out.println("Waiting for Client Request");

                                    Socket s=ss.accept();

                                    BufferedReader br;

                                    PrintStream ps;

                                    String str;

                                    br=new BufferedReader(new InputStreamReader(s.getInputStream()));

                                    str=br.readLine();

                                    System.out.println("Received radius");

                                    double r=Double.parseDouble(str);

                                    double area=3.14\*r\*r;

                                    ps=new PrintStream(s.getOutputStream());

                                    ps.println(String.valueOf(area));

                                    br.close();

                                    ps.close();

                                    s.close();

                                    ss.close();

                        }

                        catch(Exception e)

                        {

                                    System.out.println(e);

                        }

            }

}

**Output:**

Waiting for Client Request

Received radius

//Client Program

import java.io.\*;

import java.net.\*;

class Client

{

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

            {

                        Socket s=new Socket(InetAddress.getLocalHost(),1064);

                        BufferedReader br;

                        PrintStream ps;

                        String str;

                        System.out.println("Enter Radius  :");

                        br=new BufferedReader(new InputStreamReader(System.in));

                        ps=new PrintStream(s.getOutputStream());

                        ps.println(br.readLine());

                        br=new BufferedReader(new InputStreamReader(s.getInputStream()));

                        str=br.readLine();

                        System.out.println("Area of the circle is : "+str);

                        br.close();

                        ps.close();

            }

}

**Output:**

Enter Radius  :

9

Area of the circle is : 254.34

# 6a) Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No light is on when the program starts.

/Program for implementing Traffic Signals

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*<applet code="Signals" width=400 height=250></applet>\*/

public class Signals extends Applet implements ItemListener

{

            String msg="";

            Checkbox  stop,ready,go;

            CheckboxGroup cbg;

            public void init()

            {

                        cbg = new CheckboxGroup();

                        stop = new Checkbox("Stop", cbg, false);

                        ready = new Checkbox("Ready", cbg, false);

                        go= new Checkbox("Go", cbg, false);

                        add(stop);

                        add(ready);

                        add(go);

                        stop.addItemListener(this);

                        ready.addItemListener(this);

                        go.addItemListener(this);

            }

            public void itemStateChanged(ItemEvent ie)

            {

                        repaint();

            }

            public void paint(Graphics g)

            {

                        msg=cbg.getSelectedCheckbox().getLabel();

                        g.drawOval(165,40,50,50);

                        g.drawOval(165,100,50,50);

                        g.drawOval(165,160,50,50);

if(msg.equals("Stop"))

                        {

                                    g.setColor(Color.red);

                                    g.fillOval(165,40,50,50);

                        }

                        else if(msg.equals("Ready"))

                        {

                                    g.setColor(Color.yellow);

                                    g.fillOval(165,100,50,50);

                        }

                        else

                        {

                                    g.setColor(Color.green);

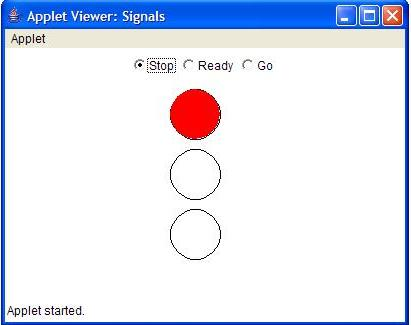
                                    g.fillOval(165,160,50,50);

                        }

            }

}

**Output:**

[](https://4.bp.blogspot.com/-8VYIlMLV6Wo/VFUV7DSHpbI/AAAAAAAAAd8/42TeZ0ikiso/s1600/1.png)

# 6b) Java applet program that allows the user to draw lines, rectangles and ovals

**import** java.awt.\*;

**import** java.applet.\*;

*/\**

*<applet code=" Rectangle " width=200 height=200>*

*</applet>*

*\*/*

**public** **class** *Rectangle* **extends** Applet

{

**public** **void** paint(Graphics g)

{

**for**(**int** i=0;i<=250;i++)

{

Color c1=**new** Color(35-i,55-i,110-i);

g.setColor(c1);

g.drawRect(250+i,250+i,100+i,100+i);

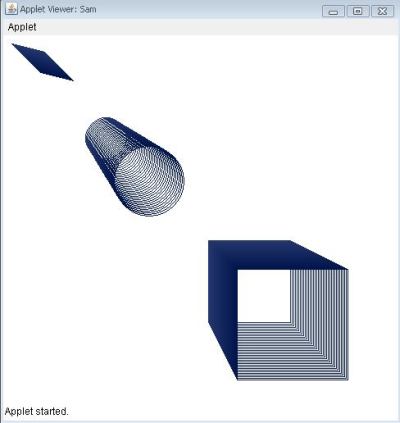
g.drawOval(100+i,100+i,50+i,50+i);

g.drawLine(50+i,20+i,10+i,10+i);

}

}

}



Output:

7a) Write a java program to create an abstract class named Shape that contains an empty method named numberofSides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberofSides ( ) that shows the number of sides in the given geometrical figures.

//Program to print number of sides for differet shapes

abstract class Shape

{

            abstract void numberofsides();

}

class Trapezoid extends Shape

{

            void numberofsides()

            {

                        System.out.println("Number of sides of Trapeziod is : 4");

            }

}

class Triangle extends Shape

{

            void numberofsides()

            {

                        System.out.println("Number of sides of Triangle is : 3");

            }

}

class Hexagon extends Shape

{

            void numberofsides()

            {

                        System.out.println("Number of sides of Hexagon is : 6");

            }

}

class Sides

{

            public static void main(String args[])

            {

                        Shape s;

                        s=new Trapezoid();

                        s.numberofsides();

                        s=new Triangle();

                        s.numberofsides();

                        s=new Hexagon();

                        s.numberofsides();

            }

}

**Output:**

Number of sides of Trapeziod is : 4

Number of sides of Triangle is : 3

Number of sides of Hexagon is : 6

7b) Suppose that a table named Table.txt is stored in a text file. The first line in the file header and the remaining lines correspond to row in the table. The elements are separated by commas. Write a Java program to display the table using labels in grid layout.

Program:-

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.\*;

import java.io.\*;

public class Table1 extends JFrame

{ int i=0; int j=0,k=0;

Object data[][]=new Object[5][4];

Object list[][]=new Object[5][4];

JButton save;

JTable table1;

FileInputStream fis;

DataInputStream dis;

public Table1()

{ String d= " ";

Container con=getContentPane();

con.setLayout(new BorderLayout());

final String[] colHeads={"Name","Roll Number","Department","Percentage"};

try

{ String s=JOptionPane.showInputDialog("Enter the File name present in the current directory");

FileInputStream fis=new FileInputStream(s);

DataInputStream dis = new DataInputStream(fis);

while ((d=dis.readLine())!=null)

{

StringTokenizer st1=new StringTokenizer(d,",");

while (st1.hasMoreTokens())

{ for (j=0;j<4;j++)

{ data[i][j]=st1.nextToken();

System.out.println(data[i][j]);

}

i++;

}

System.out.println ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

} }

catch (Exception e)

{ System.out.println ("Exception raised" +e.toString());

}

table1=new JTable(data,colHeads);

int v=ScrollPaneConstants.VERTICAL\_SCROLLBAR\_AS\_NEEDED;

int h=ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED; JScrollPane scroll=new JScrollPane(table1,v,h); con.add(scroll,BorderLayout.CENTER);

}

public static void main(String args[])

{

Table1 t=new Table1();

t.setBackground(Color.green);

t.setTitle("Display Data");

t.setSize(500,300);

t.setVisible(true);

t.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

Abc.txt:-

a,123,der,23

b,456,frg,45

Output:- screen shots to be kept

**8) Key Events**

**// demonstrate some virtual key codes.**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.applet.\*;**

**/\*<applet code="KeyEvents" width=300 height=100></applet>\*/**

**public class KeyEventsextends Applet**

**implements KeyListener**

**{**

**String msg= "";**

**int X = 10, Y = 20; // output coordinates**

**public void init()**

**{**

**addKeyListener(this);**

**}**

**public void keyPressed(KeyEventke)**

**{**

**showStatus("Key Down");**

**int key = ke.getKeyCode();**

**switch(key) {**

**case KeyEvent.VK\_PAGE\_ DOWN:**

**msg+= "<PgDn>";**

**break;**

**case KeyEvent.VK\_PAGE\_UP:**

**msg+= "<PgUp>";**

**break;**

**case KeyEvent.VK\_LEFT:**

**msg+= "<Left Arrow>";**

**break;**

**case KeyEvent.VK\_UP:**

**msg+= "<Up Arrow>";**

**break;**

**case KeyEvent.VK\_DOWN:**

**msg+= "<Down Arrow>";**

**break;**

**case KeyEvent.VK\_9:**

**msg+= "<Digit 9>";**

**break;**

**case KeyEvent.VK\_RIGHT:**

**msg+= "<Right Arrow>";**

**break;**

**}//switch**

**repaint();**

**}//keypressedmethod**

**public void keyReleased(KeyEvent ke)**

**{**

**showStatus("Key Up");**

**}**

**public void keyTyped(KeyEvent ke)**

**{**

**msg+= ke.getKeyChar();**

**repaint();**

**}**

**// Display keystrokes.**

**public void paint(Graphics g)**

**{**

**g.drawString(msg, X, Y);**

**}**

**}**

**Output:**



9) Mouse Events

// demonstrate the mouse event handlers.

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="MouseEvents" width=300

height=100></applet>\*/

public class MouseEvents extends Applet implements

MouseListener, MouseMotionListener,

MouseWheelListener

{

String msg = "";

int mouseX = 0, mouseY = 0; // coordinates of mouse

public void init() {

addMouseListener(this);

addMouseMotionListener(this);

addMouseWheelListener(this);

}

// Handle mouse clicked.

public void mouseClicked(MouseEvent me)

{

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked.";

repaint();

}

// Handle mouse entered.

public void mouseEntered(MouseEvent me)

{// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse entered.";

repaint();

}

// Handle mouse exited.

public void mouseExited(MouseEvent me)

{// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse exited.";

repaint();

}

**// Handle mouse pressed.**

**public void mousePressed(MouseEvent me) {**

**// save coordinates**

**mouseX = me.getX();**

**mouseY = me.getY();**

**msg = "Down";**

**repaint();**

**}**

**// Handle mouse released.**

**public void mouseReleased(MouseEvent me) {**

**// save coordinates**

**mouseX = me.getX();**

**mouseY = me.getY();**

**msg = "Up";**

**repaint();**

**}**

**// Handle mouse dragged.**

**public void mouseDragged(MouseEvent me)**

**{**

**// save coordinates**

**mouseX= me.getX();**

**mouseY= me.getY();**

**msg= "\*";**

**showStatus("Dragging mouse at " + mouseX+ ", " + mouseY);**

**repaint();**

**}**

**// Handle mouse moved.**

**public void mouseMoved(MouseEventme)**

**{**

**// show status**

**showStatus("Moving mouse at " + me.getX() + ", " + me.getY());**

**}**

**// Handle mouse wheel moved.**

**public void mouseWheelMoved(MouseWheelEventme) {**

**// show status**

**showStatus("Mouse Wheel Moving at " + me.getX() + ", " + me.getY());**

**}**

**// Display msgin applet window at current X,Y location.**

**public void paint(Graphics g)**

**{**

**g.drawString(msg, mouseX, mouseY);**

**}**

**}**

**Output:**



**Practice Programs**

**1) Write a program for handling window events.**

**import java.awt.\*;**

**import java.awt.event.\*;**

**class Myclass implements WindowListener**

**{**

**public void windowActivated(WindowEvent e){}**

**public void windowClosed(WindowEvent e){}**

**public void windowClosing(WindowEvent e)**

**{**

**System.exit(0);**

**}**

**public void windowDeactivated(WindowEvent e){}**

**public void windowDeiconified(WindowEvent e){}**

**public void windowIconified(WindowEvent e){}**

**public void windowOpened(WindowEvent e){}**

**}**

**class MyFrame extends Frame**

**{**

**public static void main(String[] args)**

**{**

**MyFramef = new MyFrame();**

**f.setTitle("My AWT Frame");**

**f.setSize(400,400);**

**f.setVisible(true);**

**f.addWindowListener(new Myclass());**

**}} Output:**



**2) Program to close a frame using Adapter classes**

**import java.awt.\*;**

**import java.awt.event.\*;**

**class FrameClosing extends Frame**

**{**

**FrameClosing(String title)**

**{**

**super(title);**

**MyWindowAdapter ad= new MyWindowAdapter(this);**

**addWindowListener(ad);**

**}**

**public void paint(Graphics g)**

**{**

**g.drawString("This is in frame window",60,60);**

**}**

**public static void main(String args[])**

**{**

**FrameClosingsf=new FrameClosing("FRAME CLOSING");**

**sf.setVisible(true);**

**sf.setSize(300,200);**

**}**

**}**

**class MyWindowAdapter extends WindowAdapter**

**{**

**FrameClosing sf;**

**public MyWindowAdapter(FrameClosing sf)**

**{**

**this.sf=sf;**

**}**

**public void windowClosing(WindowEvent we)**

**{**

**sf.setVisible(false);**

**System.exit(0);**

**// to exit from application System.exit(0) or Frame.dispose() can be called**

**//sf.dispose();**

**}**

**public void windowClosed(WindowEvent we)**

**{**

**sf.setVisible(false);**

**}**

**}**

**Output:**



**3)** **Program to close a frame using Swing package**

**import java.awt.\*;**

**import javax.swing.\*;**

**public class ExButton extends JFrame**

**{**

**public ExButton()**

**{**

**super("Button test");**

**Button button1 = new Button("no");**

**add(button1,BorderLayout.NORTH);**

**setSize(300,200);**

**setVisible(true);**

**}**

**public static void main(String args[])**

**{**

**ExButton eb=new ExButton();**

**eb.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**}**

**}**