CYCLE – 5

1) Program to draw Circle, Rectangle, Line in Applet.

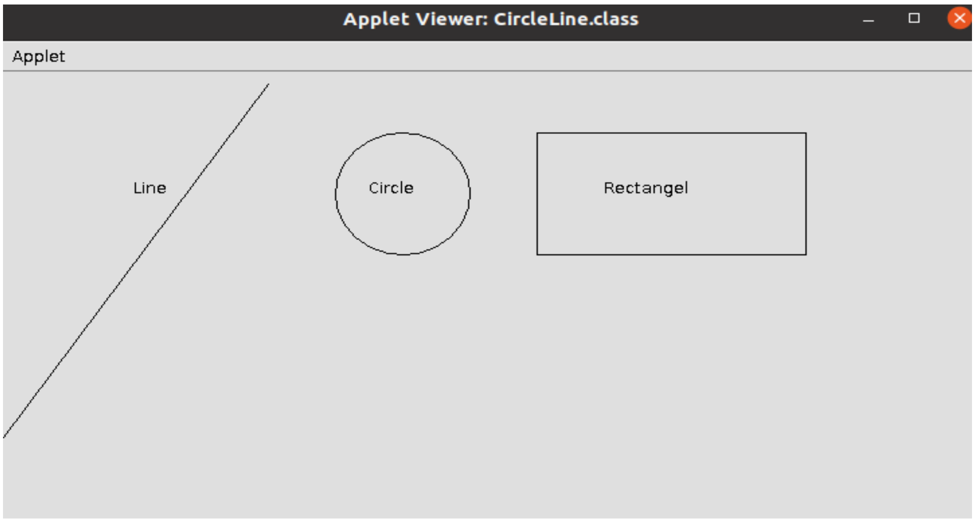
Circle_Line.java

```
import java.applet.*;
import java.awt.*;
public class  CircleLine extends Applet
{
   int x=300,y=100,r=50;
   public void paint(Graphics g)
   {
      System.out.println("Name: Ashin Siby");
      g.drawLine(3,300,200,10);
      g.drawString("Line",100,100);
      g.drawOval(x-r,y-r,100,100);
      g.drawString("Circle",275,100);
      g.drawRect(400,50,200,100);
      g.drawString("Rectangel",450,100);
  }
}
```

CircleLine_index.html

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<div align="center">
<APPLET CODE="CircleLine.class" WIDTH="800" HEIGHT="500"></APPLET>
</div>
</BODY>
</HTML>
```

Output

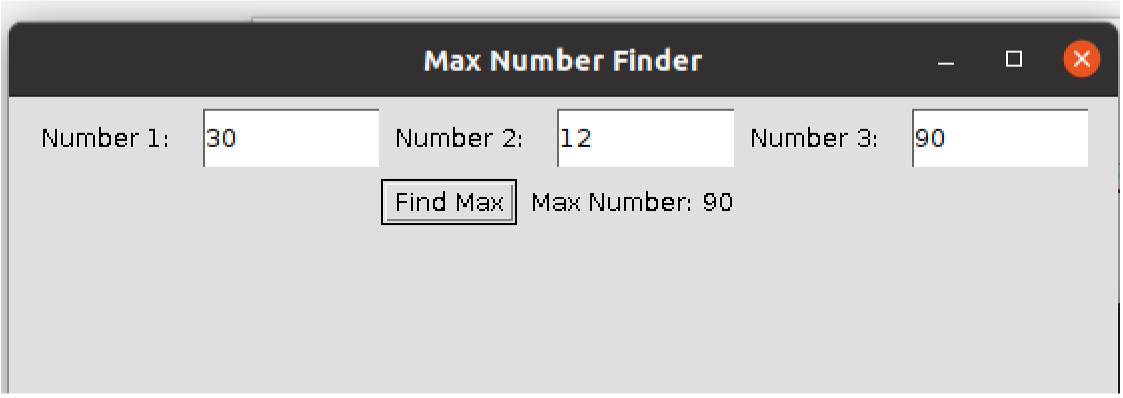


2) Program to find maximum of three numbers using AWT.

MaxNumberFinder.java

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MaxNumberFinder extends Frame {
    private TextField num1Field, num2Field, num3Field;
    private Label resultLabel;

    public MaxNumberFinder() {
        setTitle("Max Number Finder");
        setSize(300, 200);
        setLayout(new FlowLayout());

        Label num1Label = new Label("Number 1: ");
        num1Field = new TextField(10);
        add(num1Label);
        add(num1Field);

        Label num2Label = new Label("Number 2: ");
        num2Field = new TextField(10);
        add(num2Label);
        add(num2Field);

        Label num3Label = new Label("Number 3: ");
        num3Field = new TextField(10);
        add(num3Label);
        add(num3Field);

        Button findButton = new Button("Find Max");
        add(findButton);

        resultLabel = new Label("");
        add(resultLabel);

        findButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                int num1 = Integer.parseInt(num1Field.getText());
                int num2 = Integer.parseInt(num2Field.getText());
                int num3 = Integer.parseInt(num3Field.getText());

                int max = Math.max(num1, Math.max(num2, num3));

                resultLabel.setText("Max Number: " + max);
            }
        });

        setVisible(true);
    }

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

Output



Max Number Finder

Number 1: 30 Number 2: 12 Number 3: 90

Max Number: 90

3. Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

Q3.java

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

public class Q3 extends Applet implements ActionListener {
   Label l1,l2,l3,l4,l5,l6;
    TextField t1,t2,t3,t4,t5,t6;
   Button b;
   public void init(){
       l1 = new Label("MARK 1:");
        t1 = new TextField();
        l2 = new Label("MARK 2:");
        t2 = new TextField();
       l3 = new Label("MARK 3:");
        t3 = new TextField();
        l4 = new Label("MARK 4:");
        t4 = new TextField();
        l5 = new Label("MARK 5:");
        t5 = new TextField();
        l6 = new Label("PERCENTAGE:");
        t6 = new TextField();


        b = new Button("SEE STATUS");

        setLayout(null);

        l1.setBounds(450,50,70,20);
        t1.setBounds(520,50,100,20);
        l2.setBounds(450,80,70,20);
        t2.setBounds(520,80,100,20);
        l3.setBounds(450,110,70,20);
        t3.setBounds(520,110,100,20);
        l4.setBounds(450,140,70,20);
        t4.setBounds(520,140,100,20);
        l5.setBounds(450,170,70,20);
        t5.setBounds(520,170,100,20);
        l6.setBounds(450,200,100,20);
        t6.setBounds(550,200,100,20);

        b.setBounds(450,290,80,30);

        add(l1);
        add(l2);
        add(l3);
```

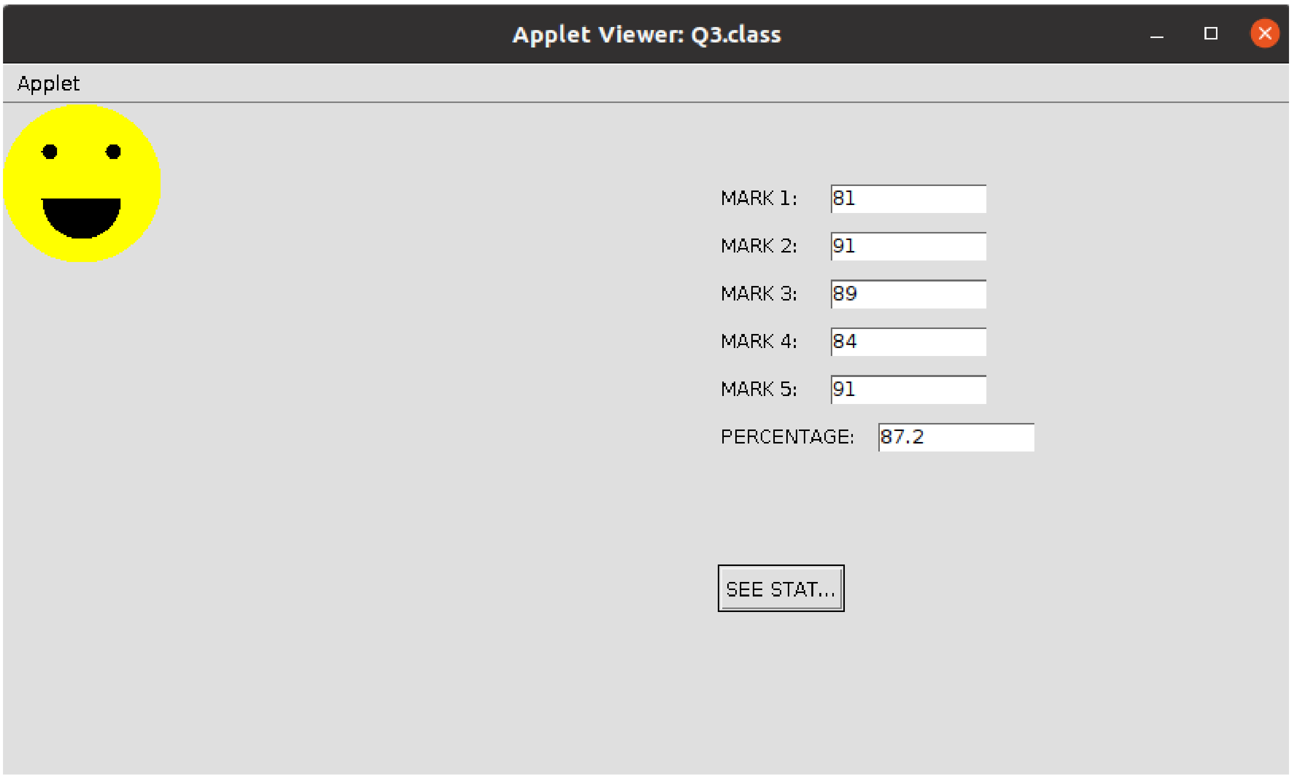
```
     add(l4);
     add(l5);
     add(l6);
     add(t1);
    add(t2);
     add(t3);
    add(t4);
     add(t5);
     add(t6);
    add(b);
    b.addActionListener(this);

}
 public void actionPerformed(ActionEvent e){
     float m1, m2,m3, m4,m5,percent;

    m1= Float.parseFloat(t1.getText());
    m2= Float.parseFloat(t2.getText());
     m3= Float.parseFloat(t3.getText());
     m4= Float.parseFloat(t4.getText());
     m5= Float.parseFloat(t5.getText());

     percent=((m1+m2+m3+m4+m5)*100)/500;

    t6.setText(String.valueOf(percent));
     repaint();
 }


public void paint(Graphics g){

     float p;
    p= Float.parseFloat(t6.getText());

     if(p> 50.0) {
         g.setColor(Color.YELLOW);
        g.fillOval(0,0,100,100);
         g.setColor(Color.black);
        g.fillOval(25,25,10,10);
       g.fillOval(65,25,10,10);
         g.setColor(Color.black);
        g.fillArc (25,35,50,50,0,-180);
     }
     else {
         g.setColor(Color.YELLOW);
         g.fillOval(0,0,100,100);
         g.setColor(Color.black);
         g.fillOval(25,25,10,10);
         g.fillOval(75,25,10,10);
        g.setColor(Color.black);
         g.drawArc(25,35,50,50,0,180);
     }
```

```
   }
}
```

Q3_index

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<div align="center">
<APPLET CODE="Q3.class" WIDTH="800" HEIGHT="500"></APPLET>
</div>
</BODY>
</HTML>
```

Output



MARK 1: 81

MARK 2: 91

MARK 3: 89

MARK 4: 84

MARK 5: 91

PERCENTAGE: 87.2

4. Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

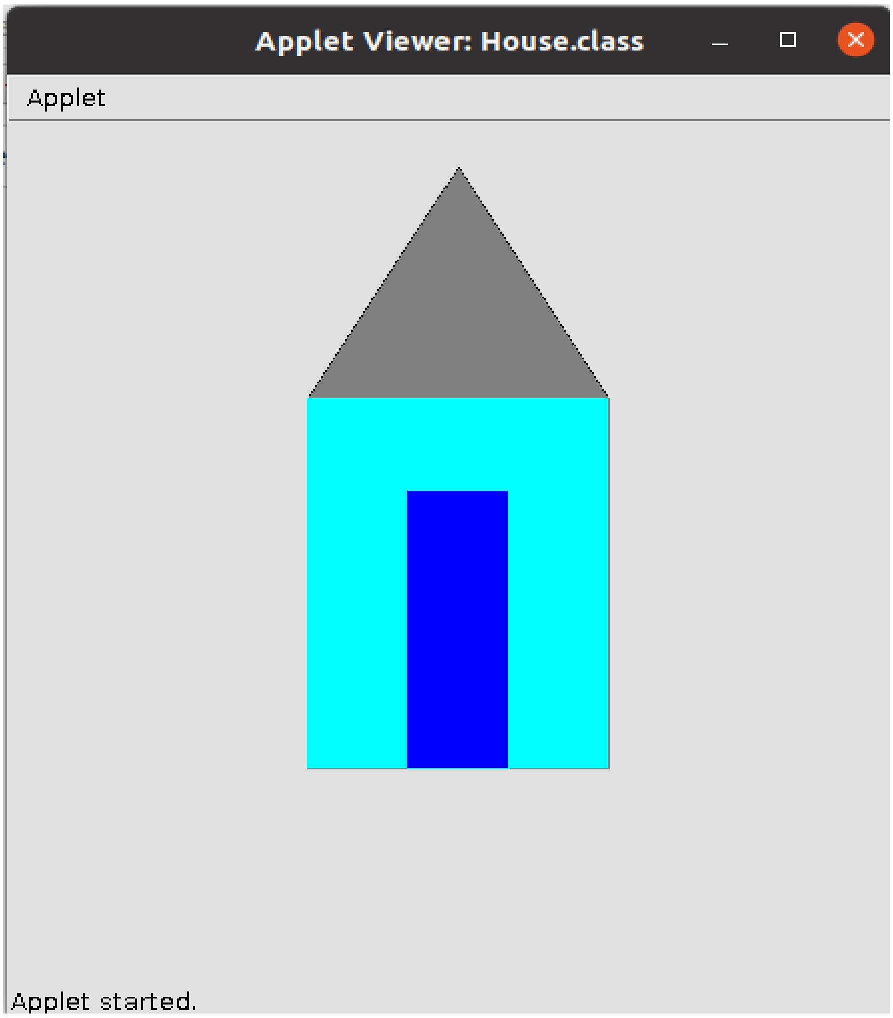
House.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class House extends Applet implements MouseListener
{
   int a,b;
    public void init()
    {
        addMouseListener( this);
    }
    public void paint(Graphics g)
    {
        int x[]={150,300,225};
        int y[]={150,150,25};
        g.drawPolygon(x,y,3);
        g.setColor(Color.GRAY);
        g.fillPolygon(x,y,3);

        g.drawRect(150,150,150,200);//House
        g.setColor(Color.CYAN);
        g.fillRect(150,150,150,200);

        g.drawRect(200,200,50,150);//Door
        g.setColor(Color.blue);
        g.fillRect(200,200,50,150);

        if(a>200 && a<300 && b>200 && b<300)
        {
            g.setColor(Color.red);
            g.fillRect(200, 200, 50, 150);
        }
    }
    public void mouseClicked(MouseEvent e)
    {
    }
    public void mouseEntered(MouseEvent e)
    {
    }

    @Override
    public void mouseExited(MouseEvent e) {
    }

   public void mousePressed(MouseEvent e)
    {
        a=e.getX();
        b=e.getY();
        repaint();
    }
   public void mouseReleased(MouseEvent e)
    {
 }
}
```

House_index.html

```
 <html>
<body>
<applet code="House.class" width="600" height="600">
</applet>
</body>
</html>
```

Output



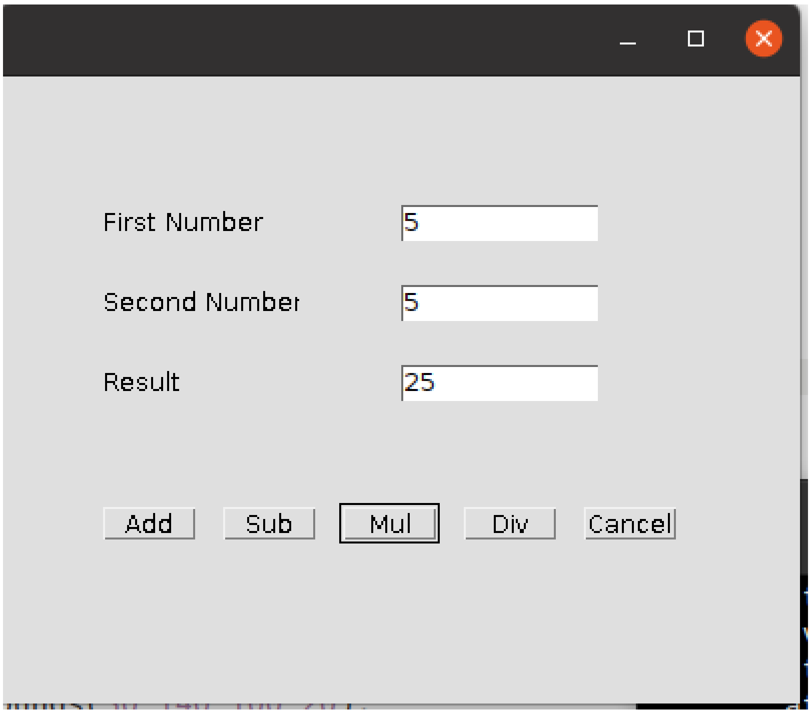
Applet started.

5. Implement a simple calculator using AWT components.

Calculator.java

```
import java.awt.*;
import java.awt.event.*;
public class Calculator implements ActionListener
{
	Frame f=new Frame();
	Label l1=new Label("First Number");
	Label l2=new Label("Second Number");
	Label l3=new Label("Result");
	TextField t1=new TextField();
	TextField t2=new TextField();
	TextField t3=new TextField();
	Button b1=new Button("Add");
	Button b2=new Button("Sub");
Button b3=new Button("Mul");
Button b4=new Button("Div");
Button b5=new Button("Cancel");
Calculator()
{
l1.setBounds(50,100,100,20);
l2.setBounds(50,140,100,20);
l3.setBounds(50,180,100,20);
t1.setBounds(200,100,100,20);
t2.setBounds(200,140,100,20);
t3.setBounds(200,180,100,20);
b1.setBounds(50,250,50,20);
b2.setBounds(110,250,50,20);
b3.setBounds(170,250,50,20);
b4.setBounds(230,250,50,20);
b5.setBounds(290,250,50,20);
f.add(l1);
f.add(l2);
f.add(l3);
f.add(t1);
f.add(t2);
f.add(t3);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
f.add(b5);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(400,350);
}
public void actionPerformed(ActionEvent e)
{
int n1=Integer.parseInt(t1.getText());
int n2=Integer.parseInt(t2.getText());
if(e.getSource()==b1)
{
t3.setText(String.valueOf(n1+n2));
}
if(e.getSource()==b2)
{
t3.setText(String.valueOf(n1-n2));
}
if(e.getSource()==b3)
{
t3.setText(String.valueOf(n1*n2));
}
if(e.getSource()==b4)
{
t3.setText(String.valueOf(n1/n2));
}
if(e.getSource()==b5)
{
System.exit(0);
}
}
public static void main(String...s)
{
new Calculator();
}
}
```

Output



First Number 5

Second Number 5

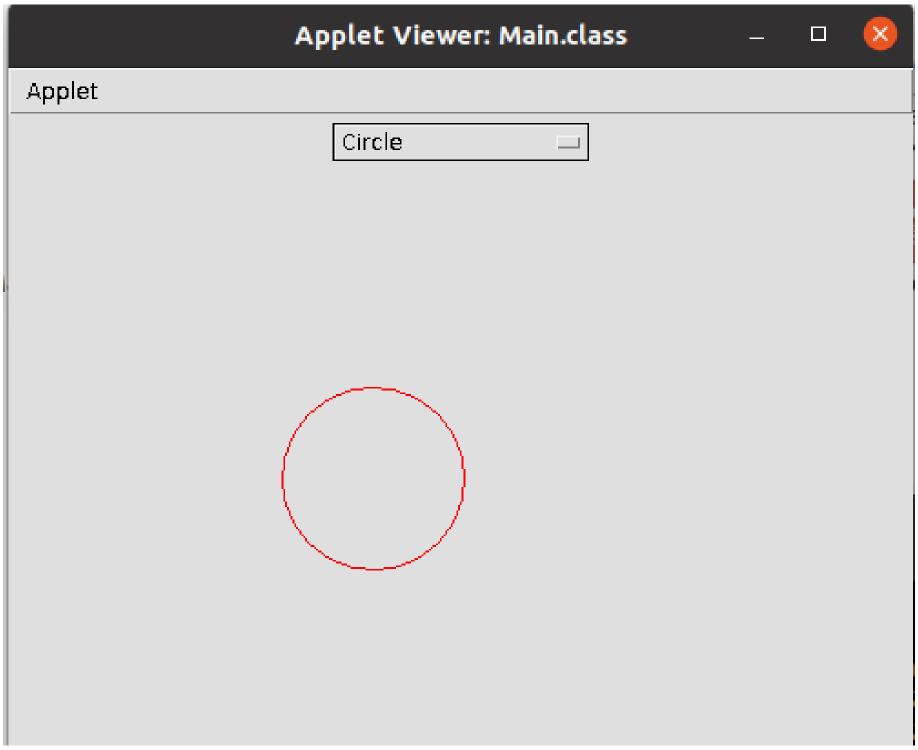
Result 25

6. Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user’s choice.

Main.java

```
import java.applet.*;
import java.awt.*;
import java.awt.Graphics;
import java.awt.event.*;
public class Main extends Applet implements ItemListener
    {
       Choice figure = new Choice();
       int Select;
       public void init()
       {
        figure.addItem("Select your choice");
        figure.addItem("Rectangle");
         figure.addItem("Square");
         figure.addItem("Circle");
         figure.addItem("Triangle");
         add(figure);
         figure.addItemListener(this);
       }
      public void itemStateChanged (ItemEvent e)
    {
        Select = figure.getSelectedIndex();
        repaint();
    }
 public void paint(Graphics g)
            {
           g.setColor(Color.red);
         super.paint(g);

         if (Select == 1)
         {

            g.drawRect(280, 100, 160,40);
         }
         if (Select == 2)
         {
             g.drawRect(50,50,100,100);
         }
        if (Select == 3)
         {

             g.drawOval(150,150,100,100);
        }
         if (Select ==4)
         {
             g.drawLine(120, 130, 280, 130);
             g.drawLine(120, 130, 200, 65);
             g.drawLine(200, 65, 280, 130);
         }
           }
}
```

Output

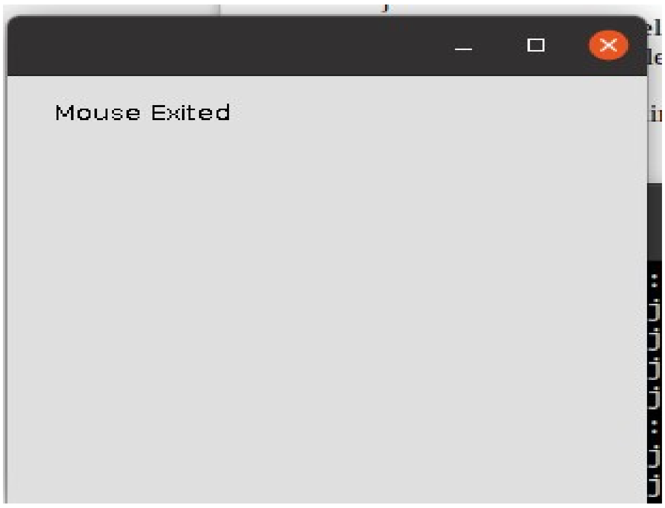


7. Develop a program to handle all mouse events and window events

Mouseevents.java

```
import java.awt.*;
import java.awt.event.*;
public class Mouseevents extends Frame implements MouseListener{
Label l;
Mouseevents(){
addMouseListener(this);
l=new Label();
l.setBounds(20,50,100,20);
add(l);
setSize(300,300);
setLayout(null);
setVisible(true);
}
public void mouseClicked(MouseEvent e) {
l.setText("Mouse Clicked");
}
public void mouseEntered(MouseEvent e) {
l.setText("Mouse Entered");
}
public void mouseExited(MouseEvent e) {
l.setText("Mouse Exited");
}
public void mousePressed(MouseEvent e) {
l.setText("Mouse Pressed");
}
public void mouseReleased(MouseEvent e) {
l.setText("Mouse Released");
}
public static void main(String[] args) {
new Mouseevents();
}
}
```

Output



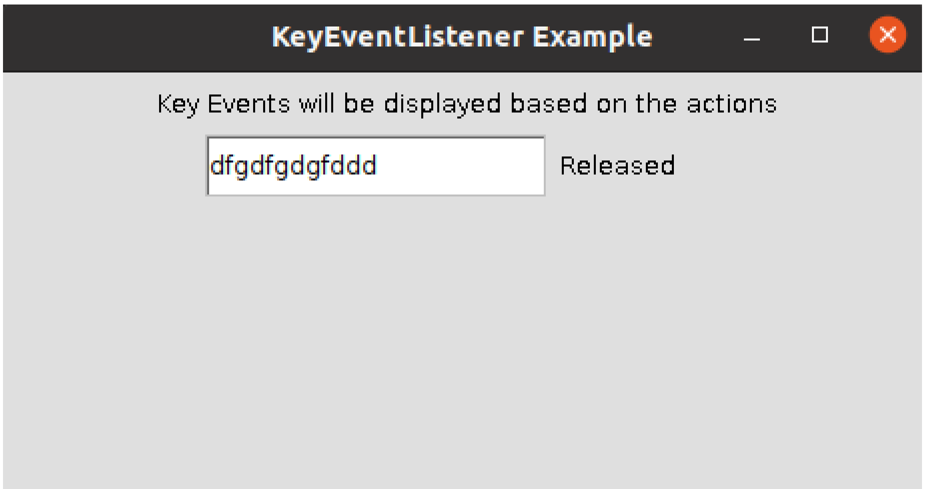
```
Mouse Exited
```

8. Develop a program to handle Key events.

KE.java

```
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
public class KE implements KeyListener
{
Label lb1, lbl2, lb;
TextField tf1;
Frame fr;
String s;
KE()
{
fr = new Frame("KeyEventListener Example");
lb1= new Label(" Key Events will be displayed based on the actions",
Label.CENTER);
lbl2= new Label();
lb= new Label();
tf1 = new TextField(20);
fr.setLayout(new FlowLayout());
fr.add(lb1);
fr.add(tf1);
fr.add(lbl2);
tf1.addKeyListener(this);
fr.setSize(460,250);
fr.setVisible(true);
}
public void keyPressed(KeyEvent ev)
{

lbl2.setText(" Key pressed");
}
public void keyReleased(KeyEvent ev)
{
lbl2.setText("Released");
}
public void keyTyped(KeyEvent ev)
{
lbl2.setText("Key is typed");
fr.setVisible(true);
}
public static void main(String[] args)
{
new KE();
}
}
```

Output



KeyEventListener Example

Key Events will be displayed based on the actions

dfgdfgdgfddd

Released