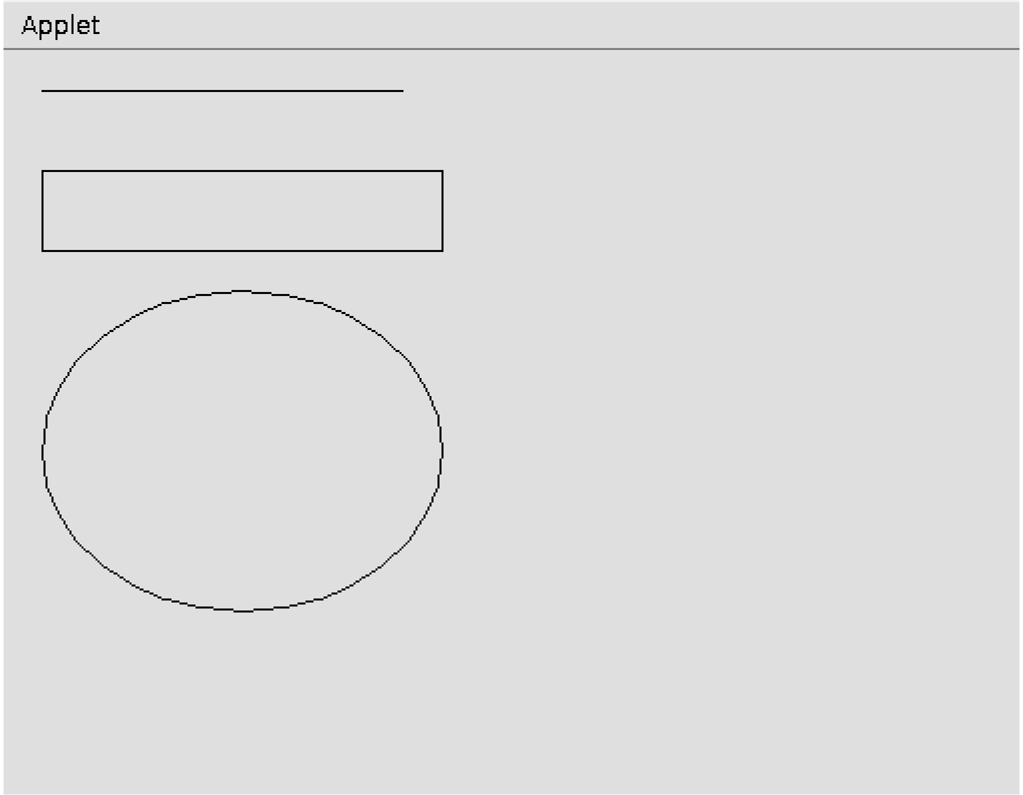
LAB CYCLE 5

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

Input:

```
import java.applet.Applet;
import java.awt.Graphics;

public class DrawingApplet extends Applet {
    public void paint(Graphics g) {
        g.drawLine(20, 20, 200, 20);
        g.drawRect(20, 60, 200, 40);
        g.drawOval(20, 120, 200, 160);
    }
}
```

Output:



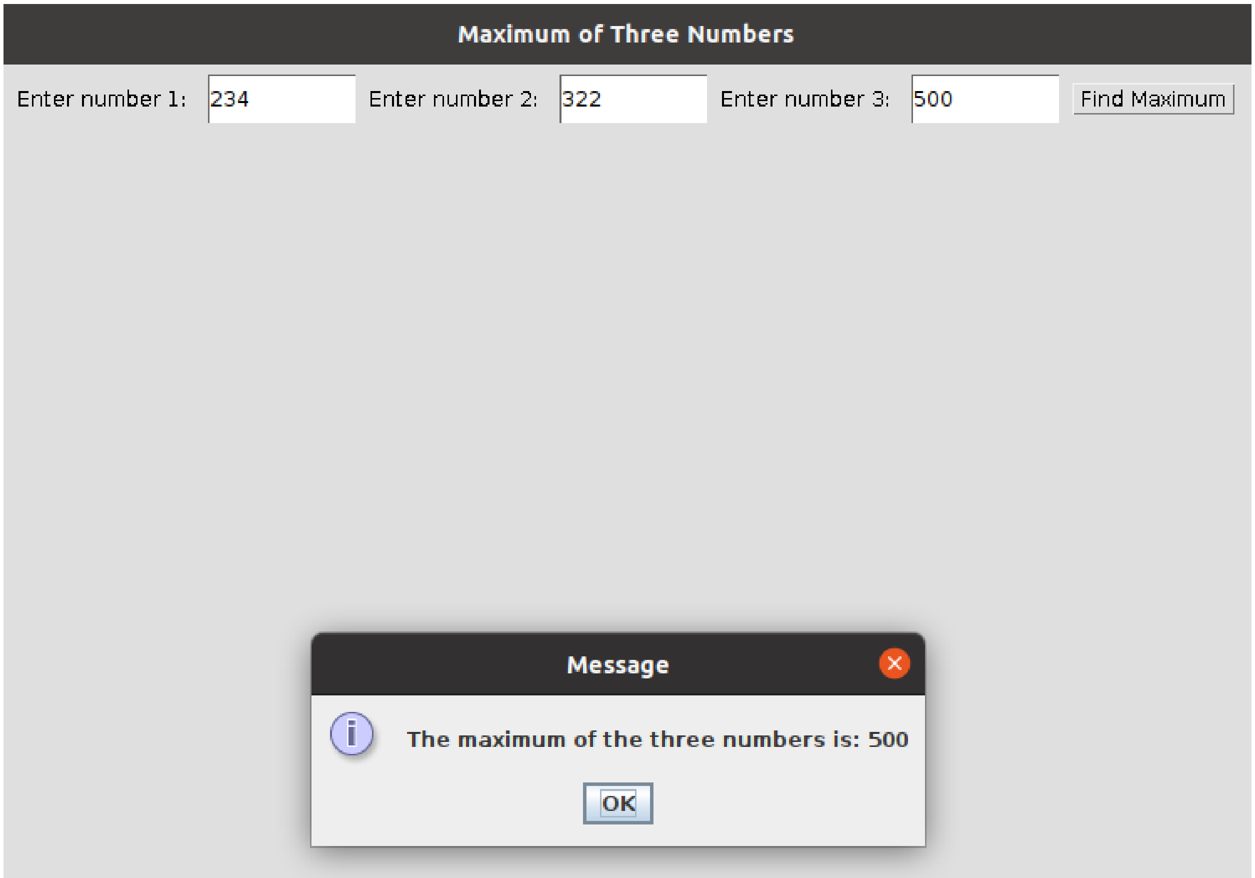
2. Program to find maximum of three numbers using AWT.

Input:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.JOptionPane;

class Max3 extends Frame implements ActionListener {
    private TextField num1Field, num2Field, num3Field;
    private Button findMaxButton;

    public Max3() {
        setTitle("Maximum of Three Numbers");
        setSize(300, 200);
        setLayout(new FlowLayout());

        Label num1Label = new Label("Enter number 1:");
        Label num2Label = new Label("Enter number 2:");
        Label num3Label = new Label("Enter number 3:");

        num1Field = new TextField(10);
        num2Field = new TextField(10);
        num3Field = new TextField(10);

        findMaxButton = new Button("Find Maximum");
        findMaxButton.addActionListener(this);

        add(num1Label);
        add(num1Field);
        add(num2Label);
        add(num2Field);
        add(num3Label);
        add(num3Field);
        add(findMaxButton);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == findMaxButton) {
            try {
                int num1 = Integer.parseInt(num1Field.getText());
                int num2 = Integer.parseInt(num2Field.getText());
                int num3 = Integer.parseInt(num3Field.getText());

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

                String message = "The maximum of the three numbers is: " +
max;
                JOptionPane.showMessageDialog(this, message);
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(this, "Please enter valid
numbers.");
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Max3 maxFinder = new Max3();
        maxFinder.setVisible(true);
    }
}
```

Output:

Output:



Maximum of Three Numbers

Enter number 1: 234

Enter number 2: 322

Enter number 3: 500

The maximum of the three numbers is: 500

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.

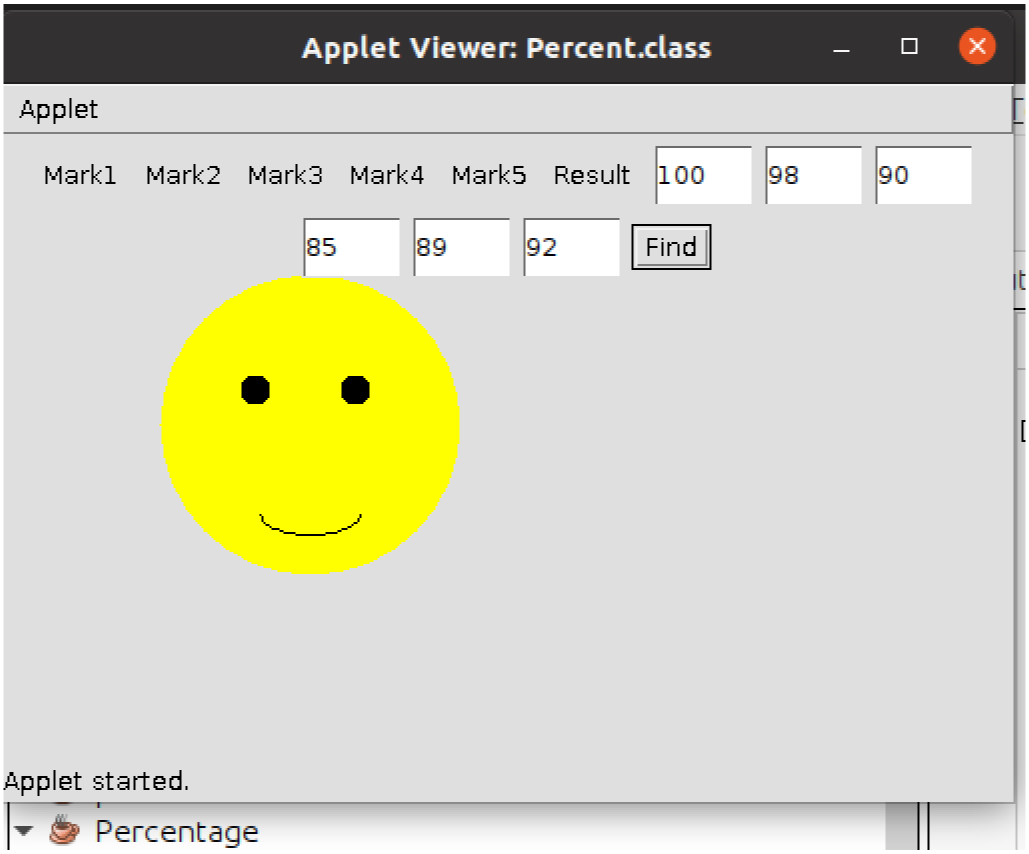
Input:

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

public class Percent extends Applet implements ActionListener {
    TextField t1,t2,t3,t4,t5,t6;
    Button b;
    Label l1,l2,l3,l4,l5,l6;
    public void init(){
        l1=new Label("Mark1");
//l1.setBounds(100,100,200,20);
        t1= new TextField(5);
//t1.setBounds(100,50,200,20);

        l2=new Label("Mark2");
//l2.setBounds(100,130,100,30);
        t2= new TextField(5);
//t2.setBounds(100,80,100,20);

        l3=new Label("Mark3");
//l3.setBounds(100,160,100,20);
        t3= new TextField(5);
//t3.setBounds(100,120,100,20);

        l4=new Label("Mark4");
//l4.setBounds(100,200,100,20);

        t4=new TextField(5);

        l5=new Label("Mark5");
        t5=new TextField(5);

        l6=new Label("Result");
        t6=new TextField(5);



        t1.setBounds(210,40,100,20);
        t2.setBounds(210,80,100,20);
        t3.setBounds(210,120,100,20);
        t4.setBounds(210,140,100,20);
        t5.setBounds(210,140,100,20);
        t6.setBounds(210,140,100,20);
        l1.setBounds(100,40,100,20);
        l2.setBounds(100,80,100,20);
        l3.setBounds(100,120,100,20);
        l4.setBounds(100,140,100,20);
        l5.setBounds(100,140,100,20);
        l6.setBounds(100,140,100,20);

        b=new Button("Find");
        b.setBounds(230,150,60,50);

//t4.setBounds(100,200,100,20);



        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){

    int x=0;
    int y=0;
    int z=0;
    int v=0;
    int w=0;
    int total=0;
    x= Integer.parseInt(t1.getText());
    y= Integer.parseInt(t2.getText());
    z= Integer.parseInt(t3.getText());
    v= Integer.parseInt(t4.getText());
    w= Integer.parseInt(t5.getText());
    if(e.getSource()==b){

        total=(x+y+z+v+w)/5;
        t6.setText(String.valueOf(total));
    }



}
@Override
public void paint(Graphics g){
    int x=0;
    int y=0;
    int z=0;
    int v=0;
    int w=0;
    int total=0;


    x= Integer.parseInt(t1.getText());
    y= Integer.parseInt(t2.getText());
    z= Integer.parseInt(t3.getText());
    v= Integer.parseInt(t4.getText());
    w= Integer.parseInt(t5.getText());

    total=(x+y+z+v+w)/5;
```

```
        if(total > 50){
           g.setColor(Color.YELLOW);
           g.fillOval(80,70, 150, 150);

           g.setColor(Color.BLACK);
           g.fillOval(120,120,15,15);
           g.fillOval(170,120,15,15);

           g.drawArc(130,180,50,20,180,180);
        }
        else
        {
           g.setColor(Color.YELLOW);
           g.fillOval(80,70, 150, 150);

           g.setColor(Color.BLACK);
           g.fillOval(120,120,15,15);
           g.fillOval(170,120,15,15);
           g.drawArc(130,180,50,20,180,-180);
        }
    }}
```

Output:



100 98 90

85 89 92

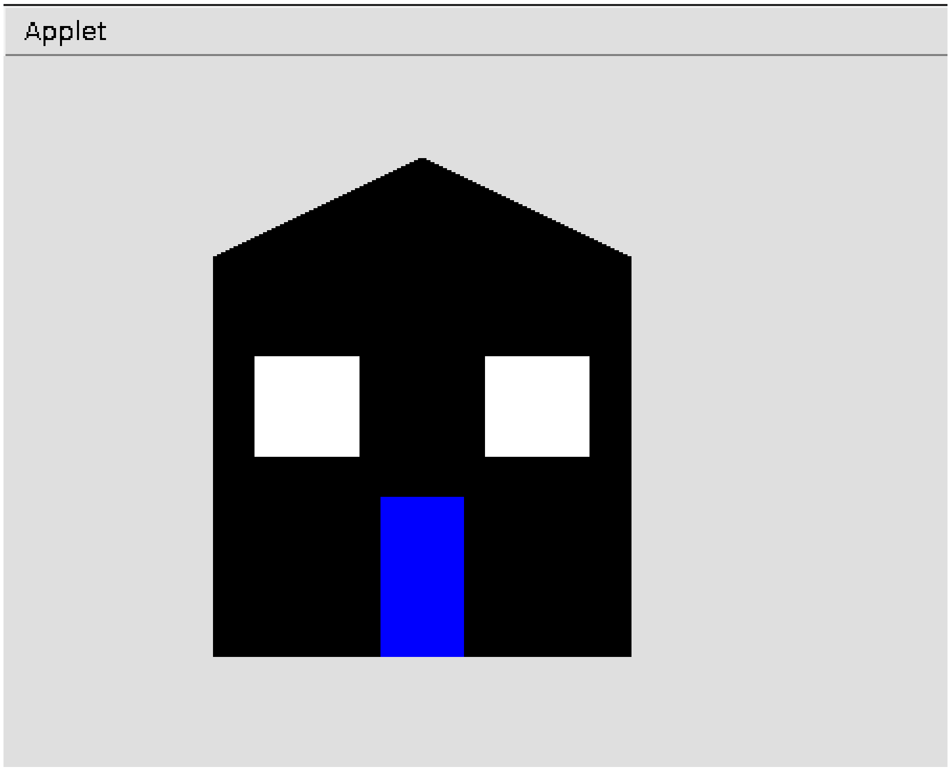
Applet started.

4. 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

Input:

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class HouseApplet extends Applet {
    private boolean doorIsBlue = true;

    @Override
    public void init() {
        addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                // Toggle door color on each mouse click
                doorIsBlue = !doorIsBlue;
                repaint(); // Redraw the applet after changing the door color
            }
        });
    }

    @Override
    public void paint(Graphics g) {
        drawHouse(g);
    }

    private void drawHouse(Graphics g) {
        // Draw the house structure
        g.setColor(Color.BLACK);
        g.fillRect(100, 100, 200, 200); // Main building
        // Draw the slanted roof
        int[] roofX = {100, 200, 300};
        int[] roofY = {100, 50, 100};
        g.fillPolygon(roofX, roofY, 3);

        // Draw the door with the selected color
        Color doorColor = doorIsBlue ? Color.BLUE : Color.RED;
        g.setColor(doorColor);
        g.fillRect(180, 220, 40, 80);

        // Draw windows
        g.setColor(Color.WHITE);
        g.fillRect(120, 150, 50, 50);
        g.fillRect(230, 150, 50, 50);
    }
}
```

Output:



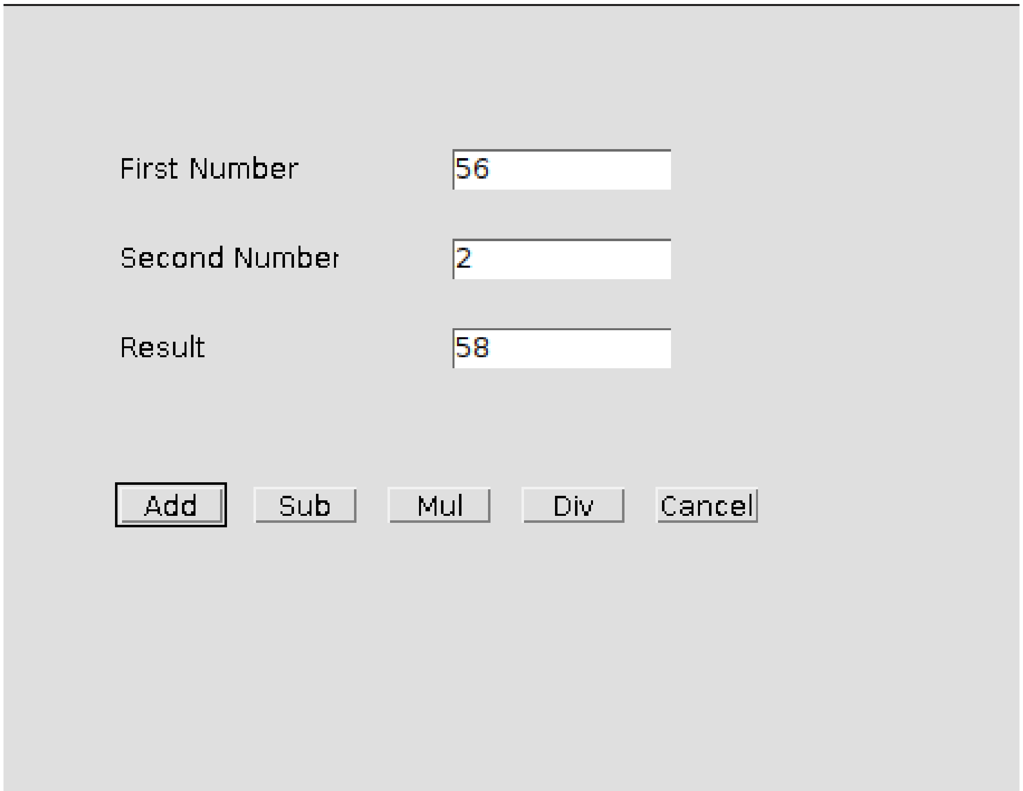
5. Implement a simple calculator using AWT components.

Input:

```
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 56

Second Number 2

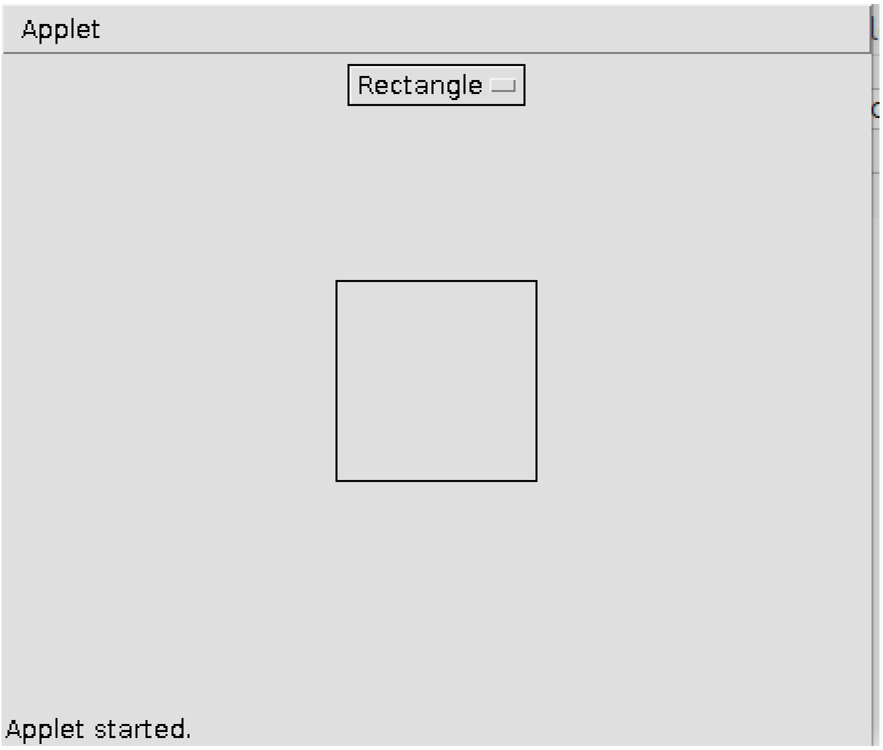
Result 58

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.

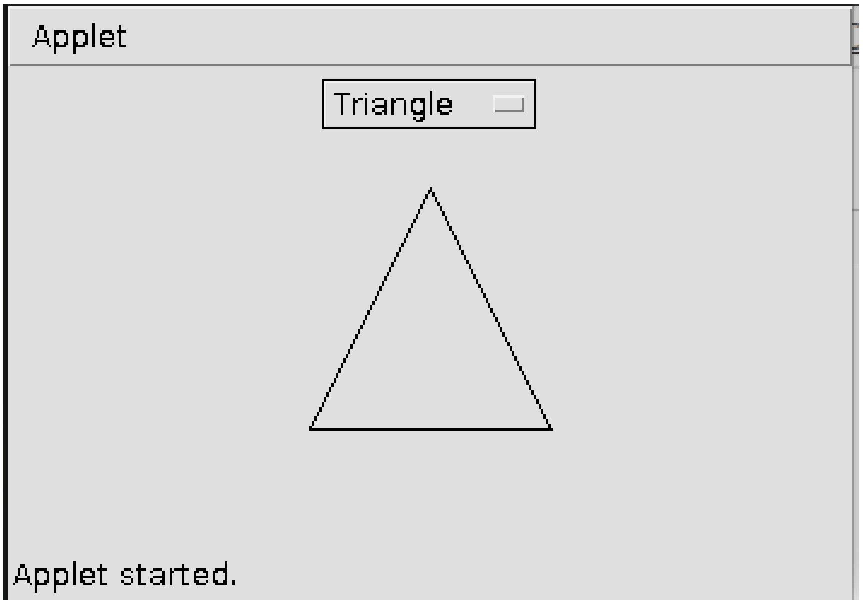
Input:

```
import java.applet.Applet;
import java.awt.Choice;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

public class ShapeName extends Applet {
    private String selectedShape = "Rectangle"; // Default shape is
Rectangle
    private int shapeWidth = 100;
    private int shapeHeight = 100;

    @Override
    public void init() {
        Choice shapeChoice = new Choice();
        shapeChoice.add("Rectangle");
        shapeChoice.add("Triangle");
        shapeChoice.add("Square");
        shapeChoice.add("Circle");

        shapeChoice.addItemListener(new ItemListener() {
            @Override
            public void itemStateChanged(ItemEvent e) {
                selectedShape = shapeChoice.getSelectedItem();
                repaint(); // Redraw the applet when the shape is changed
            }
        });

        add(shapeChoice);
    }

    @Override
    public void paint(Graphics g) {
        drawShape(g);
    }

    private void drawShape(Graphics g) {
        int centerX = getWidth() / 2;
        int centerY = getHeight() / 2;

        g.setColor(Color.BLACK);

        if (selectedShape.equals("Rectangle")) {
            g.drawRect(centerX - shapeWidth / 2, centerY - shapeHeight / 2,
shapeWidth, shapeHeight);
        } else if (selectedShape.equals("Triangle")) {
            int[] xPoints = {centerX, centerX - shapeWidth / 2, centerX +
shapeWidth / 2};
            int[] yPoints = {centerY - shapeHeight / 2, centerY + shapeHeight /
2, centerY + shapeHeight / 2};
            g.drawPolygon(xPoints, yPoints, 3);
        } else if (selectedShape.equals("Square")) {
            int sideLength = Math.min(shapeWidth, shapeHeight);
            g.drawRect(centerX - sideLength / 2, centerY - sideLength / 2,
sideLength, sideLength);
        } else if (selectedShape.equals("Circle")) {
            int diameter = Math.min(shapeWidth, shapeHeight);
            g.drawOval(centerX - diameter / 2, centerY - diameter / 2, diameter,
diameter);
        }
    }
}
```

Output:



Applet started.



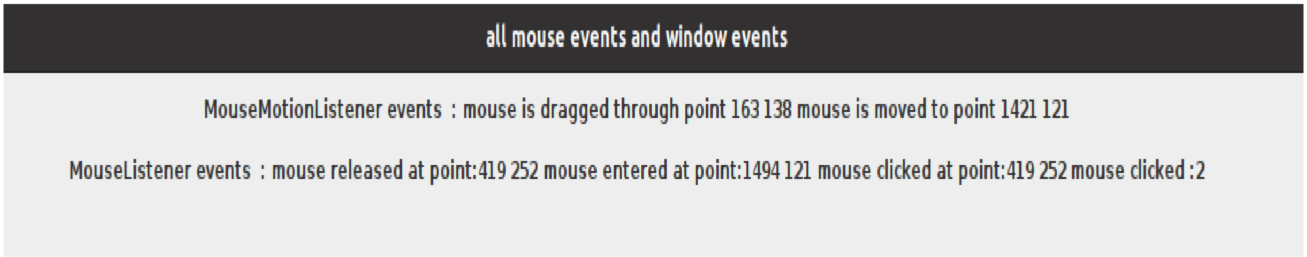
Applet started.

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

Input:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Mouse extends Frame implements MouseMotionListener,
MouseListener {

    static JLabel label1, label2, label3, label4, label5;

    Mouse() {
    }

    public static void main(String[] args) {

        JFrame f = new JFrame("all mouse events and window events");

        f.setSize(900, 300);

        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel p = new JPanel();
        JPanel p1 = new JPanel();

        f.setLayout(new FlowLayout());

        JLabel l1, l2;

        l1 = new JLabel("MouseMotionListener events  :");

        l2 = new JLabel("MouseListener events  :");

        label1 = new JLabel("no event  ");
        label2 = new JLabel("no event  ");
        label3 = new JLabel("no event  ");
        label4 = new JLabel("no event  ");
        label5 = new JLabel("no event  ");

        Mouse m = new Mouse();

        f.addMouseMotionListener(m);
        f.addMouseListener(m);

        p.add(l1);
        p.add(label1);
        p.add(label2);
        p1.add(l2);
        p1.add(label3);
        p1.add(label4);
        p1.add(label5);

        f.add(p);
        f.add(p1);

        // Corrected statement to set the frame visible
        f.setVisible(true);
    }

    public void mouseDragged(MouseEvent e) {
        label1.setText("mouse is dragged through point " + e.getX() + " " +
e.getY());
    }

    public void mouseMoved(MouseEvent e) {
        label2.setText("mouse is moved to point " + e.getX() + " " + e.getY());
    }

    public void mousePressed(MouseEvent e) {
        label3.setText("mouse pressed at point:" + e.getX() + " " + e.getY());
    }

    public void mouseReleased(MouseEvent e) {
        label3.setText("mouse released at point:" + e.getX() + " " + e.getY());
    }

    public void mouseExited(MouseEvent e) {
        label4.setText("mouse exited through point:" + e.getX() + " " +
e.getY());
    }

    public void mouseEntered(MouseEvent e) {
        label4.setText("mouse entered at point:" + e.getX() + " " + e.getY());
    }

    public void mouseClicked(MouseEvent e) {
        label5.setText("mouse clicked at point:" + e.getX() + " " + e.getY() + "
mouse clicked :" + e.getClickCount());
    }
}
```

Output:



all mouse events and window events

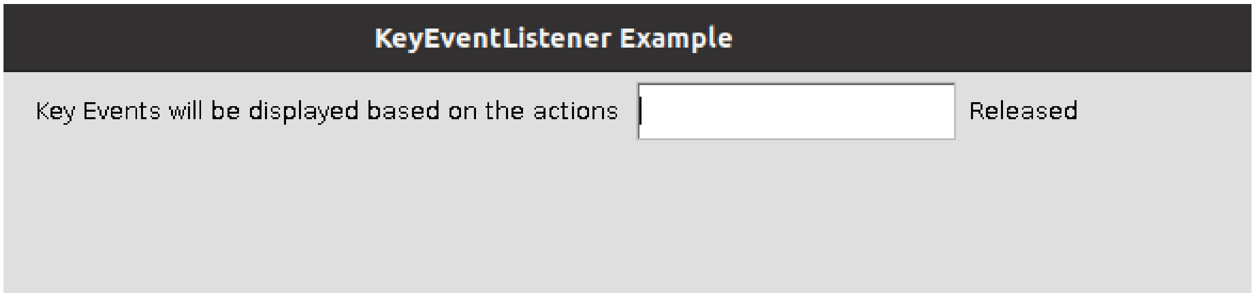
```
MouseMotionListener events  : mouse is dragged through point 163 138 mouse is moved to point 1421 121

MouseListener events  : mouse released at point:419 252 mouse entered at point:1494 121 mouse clicked at point:419 252 mouse clicked :2
```

8. Develop a program to handle Key events.

Input:

```
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 Key implements KeyListener {

    Label lb1, lbl2;
    TextField tf1;
    Frame fr;

    public Key() {
        fr = new Frame("KeyEventListener Example");

        lb1 = new Label(" Key Events will be displayed based on the actions",
Label.CENTER);
        lbl2 = 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");
    }

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

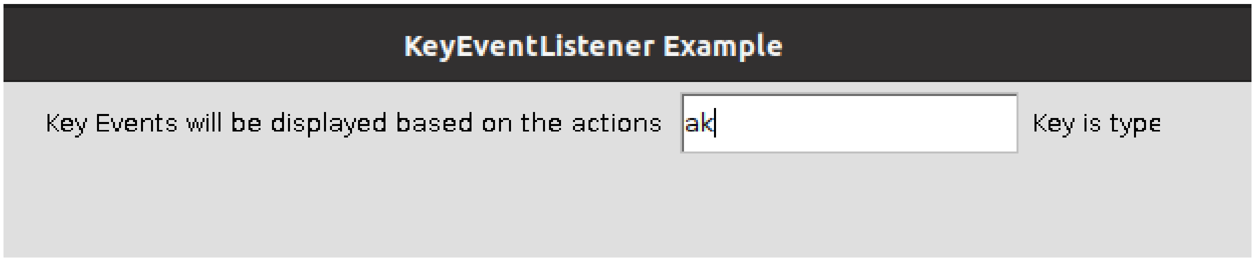
Output:



KeyEventListener Example

Key Events will be displayed based on the actions

Released



KeyEventListener Example

Key Events will be displayed based on the actions

ak

Key is type