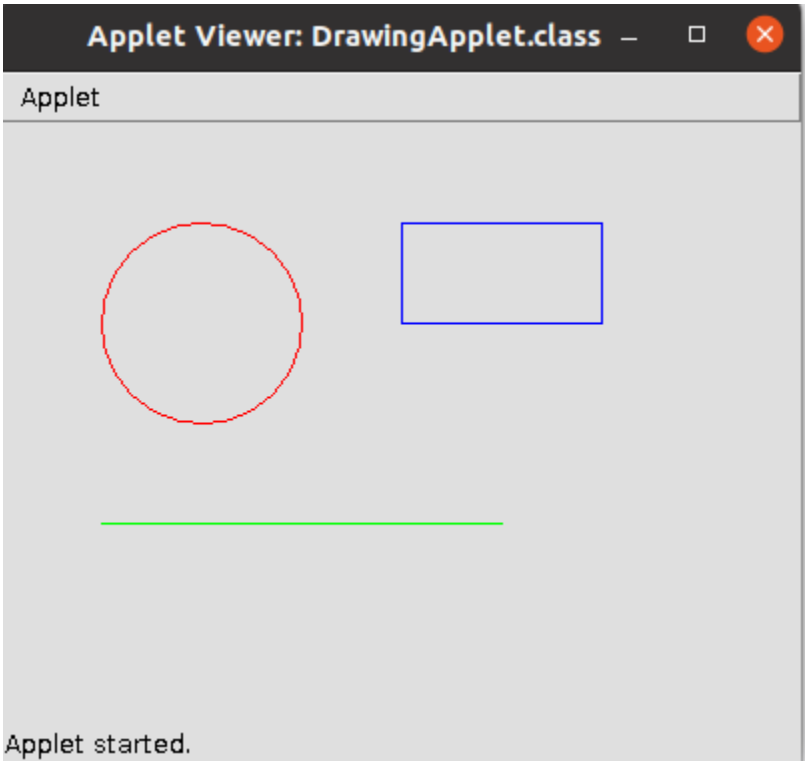
Cycle 5

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

CODE:

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

public class DrawingApplet extends Applet {
    public void paint(Graphics g) {
        // Draw a circle
        g.setColor(Color.red);
        g.drawOval(50, 50, 100, 100);

        // Draw a rectangle
        g.setColor(Color.blue);
        g.drawRect(200, 50, 100, 50);

        // Draw a line
        g.setColor(Color.green);
        g.drawLine(50, 200, 250, 200);
    }
}
/*
<!DOCTYPE html>
<html>
<head>
    <title>Java Applet Drawing</title>
</head>
<body>
    <applet code="DrawingApplet.class" width="400" height="300"></applet>
</body>
</html>
*/
```

OUTPUT:



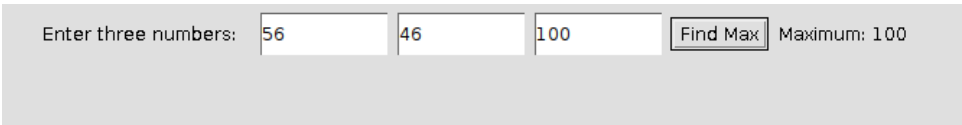
2. Program to find maximum of three numbers using AWT.

CODE:

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

public class MaxOfThreeNumbers extends Frame implements ActionListener {
    private TextField numField1, numField2, numField3;
    private Label resultLabel;

    public MaxOfThreeNumbers() {
        // Set Frame properties
        setTitle("Max of Three Numbers");
        setSize(300, 200);
        setLayout(new FlowLayout());
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });

        // Create TextFields for number input
        numField1 = new TextField(10);
        numField2 = new TextField(10);
        numField3 = new TextField(10);

        // Create Button
        Button findButton = new Button("Find Max");
        findButton.addActionListener(this);

        // Create Label for displaying the result
        resultLabel = new Label("Result will be shown here.");

        // Add components to the Frame
```

```
        add(new Label("Enter three numbers: "));
        add(numField1);
        add(numField2);
        add(numField3);
        add(findButton);
        add(resultLabel);
    }

    public void actionPerformed(ActionEvent ae) {
        // Get the numbers from TextFields
        int num1 = Integer.parseInt(numField1.getText());
        int num2 = Integer.parseInt(numField2.getText());
        int num3 = Integer.parseInt(numField3.getText());

        // Find the maximum of the three numbers
        int max = Math.max(Math.max(num1, num2), num3);

        // Display the result in the Label
        resultLabel.setText("Maximum: " + max);
    }

    public static void main(String[] args) {
        MaxOfThreeNumbers app = new MaxOfThreeNumbers();
        app.setVisible(true);
    }
}
```

OUTPUT:



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.

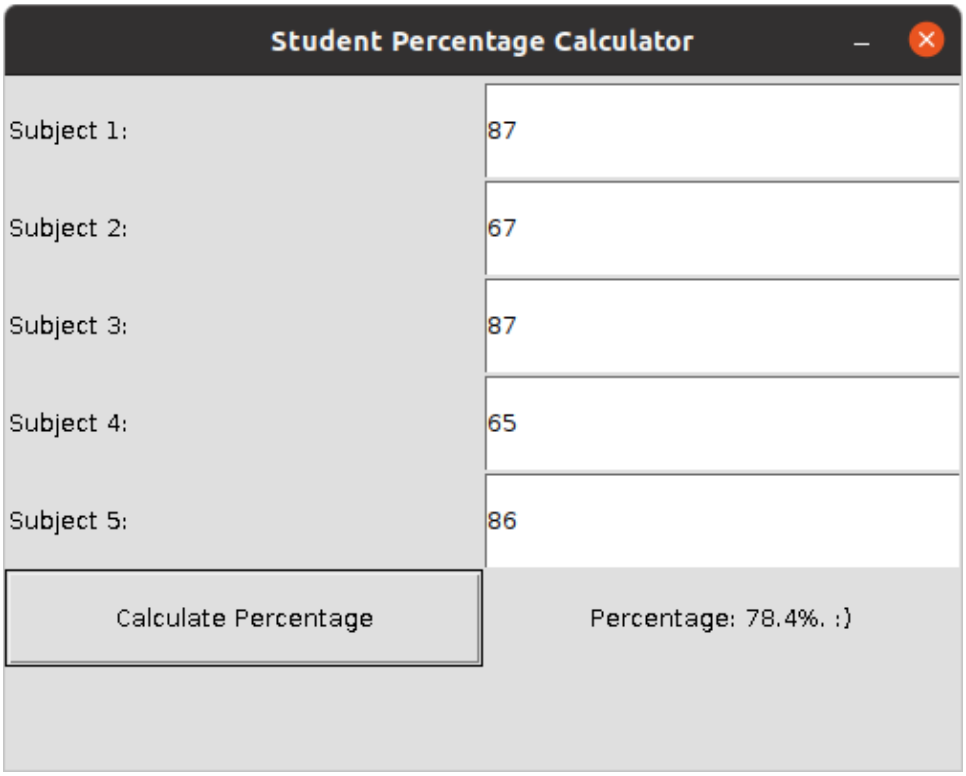
CODE:

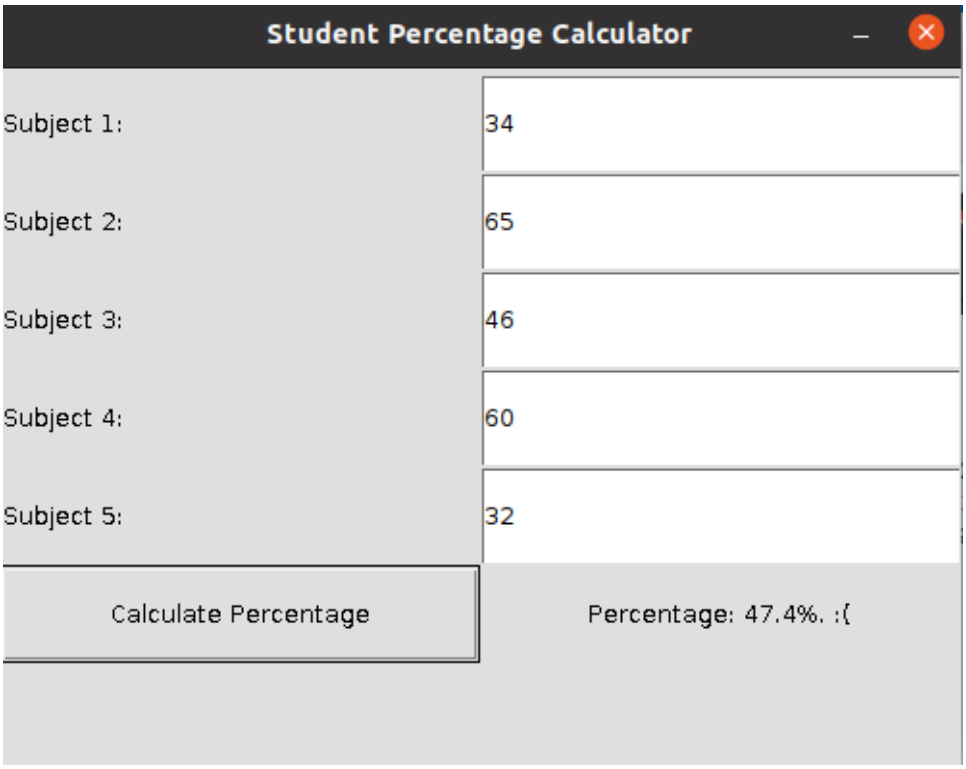
```
import java.awt.*;
import java.awt.event.*;

public class AWTStudentPercentage extends Frame implements ActionListener {
    private Label[] subjectLabels;
    private TextField[] marksFields;
    private Button calculateButton;
    private Label resultLabel;

    public AWTStudentPercentage() {
        setTitle("Student Percentage Calculator");
        setLayout(new GridLayout(7, 2));
        setSize(500, 400);
        setVisible(true);
        setResizable(false);

        subjectLabels = new Label[5];
        marksFields = new TextField[5];

        for (int i = 0; i < 5; i++) {
            subjectLabels[i] = new Label("Subject " + (i + 1) + ":");
            marksFields[i] = new TextField(5);
            add(subjectLabels[i]);
            add(marksFields[i]);
        }

        calculateButton = new Button("Calculate Percentage");
        calculateButton.addActionListener(this);
        add(calculateButton);
```

```
        resultLabel = new Label("");
        resultLabel.setAlignment(Label.CENTER);
        add(resultLabel);

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

    public void actionPerformed(ActionEvent e) {
        int totalMarks = 0;
        for (int i = 0; i < 5; i++) {
            try {
                int marks = Integer.parseInt(marksFields[i].getText());
                totalMarks += marks;
            } catch (NumberFormatException ex) {
                resultLabel.setText("Please enter valid integer marks.");
                return;
            }
        }

        double percentage = (totalMarks / 500.0) * 100.0;

        if (percentage > 50) {
            resultLabel.setText("Percentage: " + percentage + "%. :)");
        } else {
            resultLabel.setText("Percentage: " + percentage + "%. :(");
        }
    }

    public static void main(String[] args) {
```

```
        new AWTStudentPercentage();
    }
}
```

OUTPUT:




Student Percentage Calculator
Subject 1:
34
Subject 2:
65
Subject 3:
46
Subject 4:
60
Subject 5:
32
Calculate Percentage
Percentage: 47.4%. :(

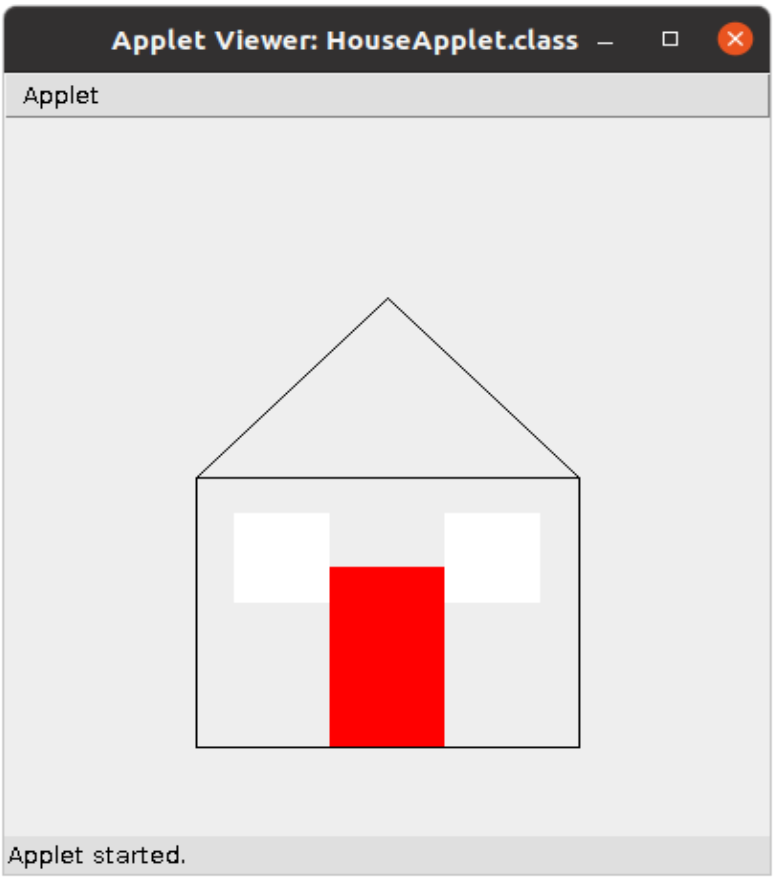
4. Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

CODE:

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

public class HouseApplet extends JApplet implements MouseListener {

    private Color doorColor = Color.BLUE;

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

    public void paint(Graphics g) {
        super.paint(g);

        // Draw the house
        g.setColor(Color.BLACK);
        g.drawRect(100, 200, 200, 150); // House body
        g.drawLine(100, 200, 200, 100); // Left roof
        g.drawLine(200, 100, 300, 200); // Right roof

        // Draw the door with the current doorColor
        g.setColor(doorColor);
        g.fillRect(170, 250, 60, 100); // Door

        // Draw the windows
        g.setColor(Color.WHITE);
        g.fillRect(120, 220, 50, 50); // Window 1
        g.fillRect(230, 220, 50, 50); // Window 2
```

```
    }

    // Mouse click event handler
    public void mouseClicked(MouseEvent e) {
        if (doorColor == Color.BLUE) {
            doorColor = Color.RED;
        } else {
            doorColor = Color.BLUE;
        }
        repaint(); // Redraw the applet to reflect the new color
    }

    // Other mouse event handlers (empty for this example)
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
}
/*
<!DOCTYPE html>
<html>
<head>
    <title>House Applet</title>
</head>
<body>
    <applet code="HouseApplet.class" width="400" height="400">
        Your browser does not support the <code>applet</code> tag.
    </applet>
</body>
</html>
*/
```

OUTPUT:


Applet Viewer: HouseApplet.class
Applet
Applet started.

5. Implement a simple calculator using AWT components.

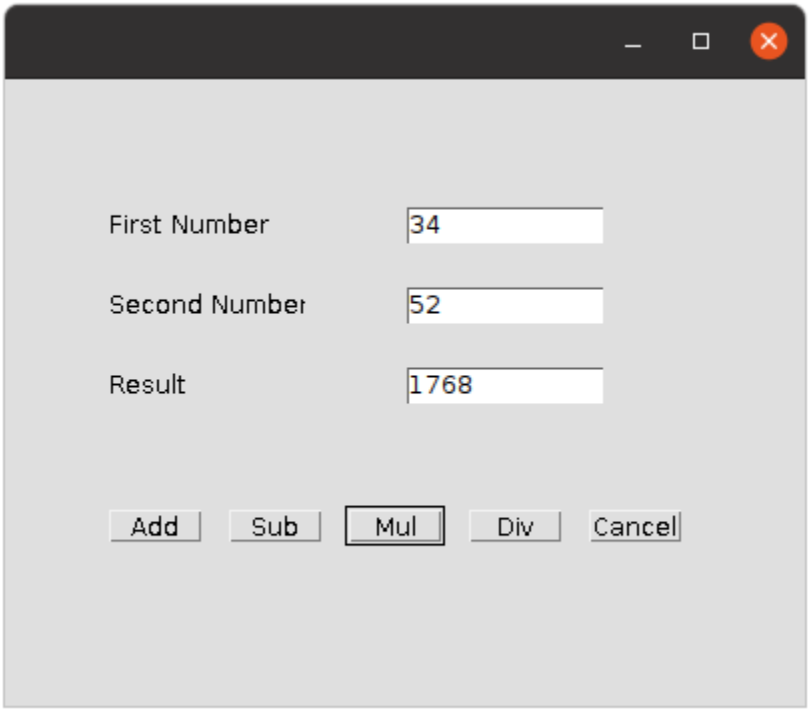
CODE:

```
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:



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.

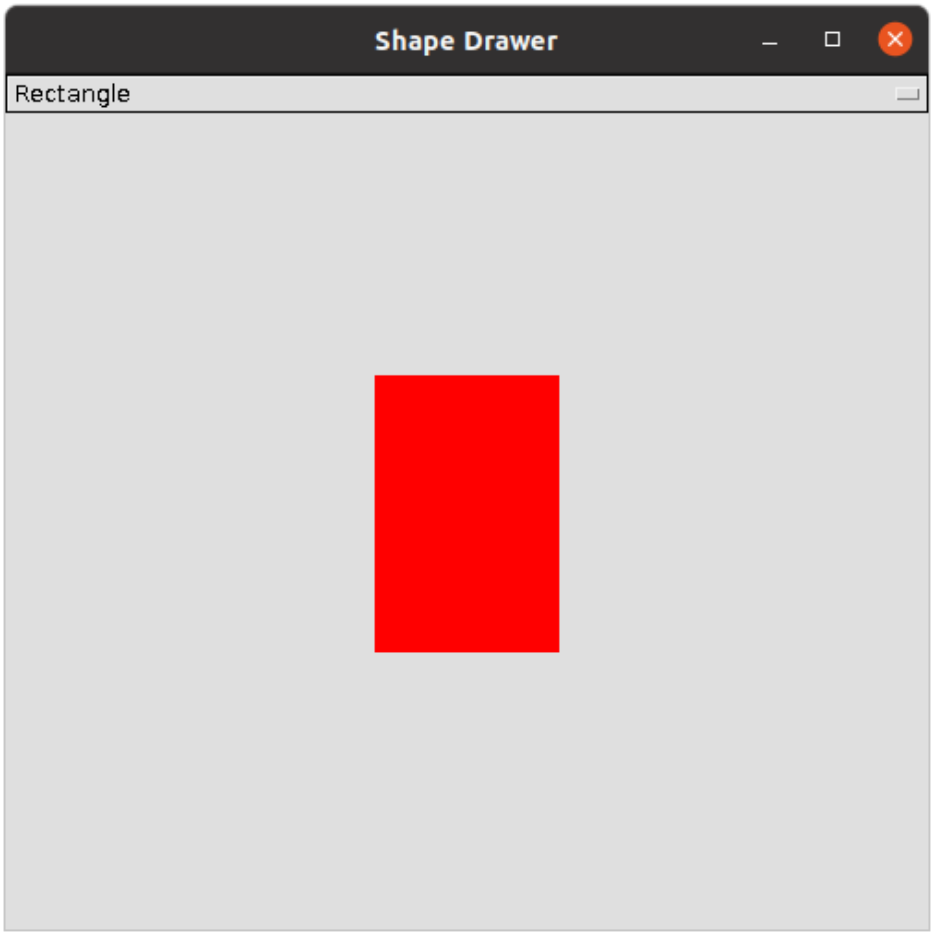
CODE:

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

public class ShapeDrawer extends Frame implements ItemListener {
    private Choice shapeChoice;

    private String selectedShape = "";

    public ShapeDrawer() {
        setTitle("Shape Drawer");
        setSize(500, 500);

        shapeChoice = new Choice();
        shapeChoice.add("Select Shape");
        shapeChoice.add("Rectangle");
        shapeChoice.add("Triangle");
        shapeChoice.add("Square");
        shapeChoice.add("Circle");
        shapeChoice.addItemListener(this);

        add(shapeChoice, BorderLayout.NORTH);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent windowEvent) {
                System.exit(0);
            }
        });
```

```
        setVisible(true);
    }

    @Override
    public void itemStateChanged(ItemEvent e) {
        selectedShape = shapeChoice.getSelectedItem();
        repaint();
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        int width = 100;
        int height = 150;

        if (selectedShape.equals("Rectangle")) {
            g.setColor(Color.RED);
            g.fillRect(200, 200, width, height);
        } else if (selectedShape.equals("Triangle")) {
            int[] xPoints = {300, 200, 400};
            int[] yPoints = {200, 300, 300};
            g.setColor(Color.BLUE);
            g.fillPolygon(xPoints, yPoints, 3);
        } else if (selectedShape.equals("Square")) {
            g.setColor(Color.GREEN);
            g.fillRect(200, 200, width, width);
        } else if (selectedShape.equals("Circle")) {
            g.setColor(Color.ORANGE);
            g.fillOval(200, 200, 100, 100);
        }
    }

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

```
  }
}
```

OUTPUT:



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

CODE:

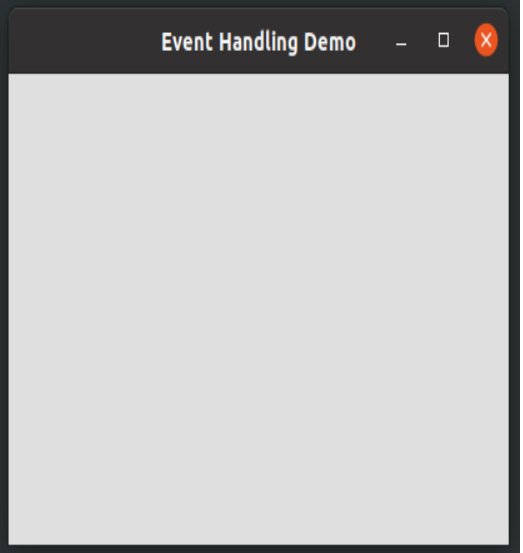
```
import java.awt.*;
import java.awt.event.*;

public class EventHandlingDemo extends Frame implements MouseListener, WindowListener {
    public EventHandlingDemo() {
        setTitle("Event Handling Demo");
        setSize(400, 300);

        addMouseListener(this);
        addWindowListener(this);

        setVisible(true);
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        System.out.println("Mouse Clicked at: " + e.getX() + ", " + e.getY());
    }

    @Override
    public void mousePressed(MouseEvent e) {
        System.out.println("Mouse Pressed at: " + e.getX() + ", " + e.getY());
    }

    @Override
    public void mouseReleased(MouseEvent e) {
        System.out.println("Mouse Released at: " + e.getX() + ", " + e.getY());
    }
```

```
@Override
public void mouseEntered(MouseEvent e) {
    System.out.println("Mouse Entered at: " + e.getX() + ", " + e.getY());
}

@Override
public void mouseExited(MouseEvent e) {
    System.out.println("Mouse Exited at: " + e.getX() + ", " + e.getY());
}

@Override
public void windowOpened(WindowEvent e) {
    System.out.println("Window Opened");
}

@Override
public void windowClosing(WindowEvent e) {
    System.out.println("Window Closing");
    System.exit(0);
}

@Override
public void windowClosed(WindowEvent e) {
    System.out.println("Window Closed");
}

@Override
public void windowIconified(WindowEvent e) {
    System.out.println("Window Iconified");
}

@Override
public void windowDeiconified(WindowEvent e) {
    System.out.println("Window Deiconified");
```

```
    }

    @Override
    public void windowActivated(WindowEvent e) {
        System.out.println("Window Activated");
    }

    @Override
    public void windowDeactivated(WindowEvent e) {
        System.out.println("Window Deactivated");
    }

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

OUTPUT:

```
sjcet@Z238-UL:~/christeenajoy/java/cycle 5/q7$ java EventHandlingDemo
Window Opened
Window Activated
Mouse Entered at: 238, 38
Mouse Exited at: 165, 30
Mouse Entered at: 304, 43
Mouse Exited at: 388, 35
Mouse Entered at: 165, 50
Mouse Pressed at: 204, 133
Mouse Released at: 204, 133
Mouse Clicked at: 204, 133
Mouse Pressed at: 121, 173
Mouse Released at: 121, 173
Mouse Clicked at: 121, 173
Mouse Pressed at: 308, 206
Mouse Released at: 308, 206
Mouse Clicked at: 308, 206
Mouse Exited at: 301, 310
Mouse Entered at: 10, 243
Mouse Exited at: 169, 35
Mouse Entered at: 184, 42
Mouse Exited at: 261, 31
Mouse Entered at: 167, 46
Mouse Exited at: 17, 36
Mouse Entered at: 111, 42
Mouse Pressed at: 353, 216
Mouse Released at: 353, 216
Mouse Clicked at: 353, 216
Mouse Exited at: 406, 253
Window Deactivated
Window Activated
Mouse Entered at: 4, 258
```



8. Develop a program to handle Key events.

CODE:

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

public class KeyEventHandlingDemo extends Frame implements KeyListener {
    private Label resultLabel;

    public KeyEventHandlingDemo() {
        setTitle("Key Event Handling Demo");
        setSize(400, 300);

        resultLabel = new Label("Press any key...");
        resultLabel.setAlignment(Label.CENTER);
        add(resultLabel);

        addKeyListener(this);

        setVisible(true);
    }

    @Override
    public void keyTyped(KeyEvent e) {
        char keyChar = e.getKeyChar();
        resultLabel.setText("Key Typed: " + keyChar);
    }

    @Override
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        resultLabel.setText("Key Pressed: " + KeyEvent.getKeyText(keyCode));
    }
```

```
    @Override
    public void keyReleased(KeyEvent e) {
        int keyCode = e.getKeyCode();
        resultLabel.setText("Key Released: " + KeyEvent.getKeyText(keyCode));
    }

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

OUTPUT: