

Experiment 1

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

public class KeyEventDemo extends JFrame implements KeyListener {
    String msg = "";

    public KeyEventDemo() {
        setTitle("Key Event Demo");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        addKeyListener(this);
        setFocusable(true);
        setVisible(true);
    }

    public void keyReleased(KeyEvent k) {
        msg = "Key Released";
        repaint();
    }

    public void keyTyped(KeyEvent k) {
        msg = "Key Typed";
        repaint();
    }

    public void keyPressed(KeyEvent k) {
        msg = "Key Pressed";
        repaint();
    }

    public void paint(Graphics g) {
        super.paint(g);
        g.drawString(msg, 100, 100);
    }

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

Experiment 2

```
import java.awt.*;
import java.awt.event.*;
public class Main extends Frame implements MouseListener {
    Label l;
    Main() {
        super("AWT Frame");

        l = new Label();
        l.setBounds(25, 60, 250, 30);
        l.setAlignment(Label.LEFT);
        this.add(l);
        this.setSize(300, 300);
        this.setLayout(null);
        this.setVisible(true);
        this.addMouseListener(this);
        this.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args) {
        new Main();
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        l.setText("Mouse Clicked");
    }
    @Override
    public void mousePressed(MouseEvent e) { }
    @Override
    public void mouseReleased(MouseEvent e) { }
    @Override
    public void mouseEntered(MouseEvent e) {
        l.setText("Mouse Entered");
    }

    @Override
    public void mouseExited(MouseEvent e) {
        l.setText("Mouse Exited");
    }
}
```

Experiment 3

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

public class StudentForm {
    // Function to write a student information in JFrame and storing it in a file
    public static void StudentInfo()
    {
        // Creating a new frame using JFrame
        JFrame f = new JFrame("Student Registration Form");
        // Creating the labels
        JLabel l1, l2, l3, l4, l5;
        // Creating three text fields for student name, college mail ID and for Mobile No
        JTextField t1, t2, t3;
        // Creating two JComboboxes for Branch and for Section
        JComboBox j1, j2;
        // Creating two buttons
        JButton b1, b2;
        // Naming the labels and setting
        // the bounds for the labels
        l1 = new JLabel("Student Name:");
        l1.setBounds(50, 50, 100, 30);
        l2 = new JLabel("College Email ID:");
        l2.setBounds(50, 120, 120, 30);
        l3 = new JLabel("Branch:");
        l3.setBounds(50, 190, 50, 30);
        l4 = new JLabel("Section:");
        l4.setBounds(420, 50, 70, 30);
        l5 = new JLabel("Mobile No:");
        l5.setBounds(420, 120, 70, 30);
        // Creating the textfields and
        // setting the bounds for textfields
        t1 = new JTextField();
        t1.setBounds(150, 50, 130, 30);
        t2 = new JTextField();
        t2.setBounds(160, 120, 130, 30);
        t3 = new JTextField();
        t3.setBounds(490, 120, 130, 30);
        // Creating two string arrays one for
        // branches and other for sections
        String s1[]
            = { " ", "CSE", "ECE", "EEE",
              "CIVIL", "MEC", "Others" };
        String s2[]
            = { " ", "Section-A", "Section-B",
              "Section-C", "Section-D",
```

```

        "Section-E" };
    // Creating two JComboBoxes for selecting branch and other for selecting the section
    // and setting the bounds
    j1 = new JComboBox(s1);
    j1.setBounds(120, 190, 100, 30);
    j2 = new JComboBox(s2);
    j2.setBounds(470, 50, 140, 30);
    // Creating one button for Saving and other button to close
    // and setting the bounds
    b1 = new JButton("Save");
    b1.setBounds(150, 300, 70, 30);
    b2 = new JButton("close");
    b2.setBounds(420, 300, 70, 30);
    // Adding action listener
    b1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
            // Getting the text from text fields
            // and JComboboxes
            // and copying it to a strings
            String s1 = t1.getText();
            String s2 = t2.getText();
            String s3 = j1.getSelectedItem() + "";
            String s4 = j2.getSelectedItem() + "";
            String s5 = t3.getText();
            if (e.getSource() == b1) {
                try {
                    // Creating a file and writing the data into a Textfile.
                    FileWriter w = new FileWriter("data.txt", true);

                    w.write(s1 + "\n");

                    w.write(s2 + "\n");
                    w.write(s3 + "\n");
                    w.write(s4 + "\n");
                    w.write(s5 + "\n");
                    w.close();
                }
                catch (Exception ae) {
                    System.out.println(ae);
                }
            }
            // Shows a Pop up Message when
            // save button is clicked
            JOptionPane.showMessageDialog(f, "Successfully Saved"+ " The Details");
        }
    });
    // Action listener to close the form

```

```

b2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)

{
    f.dispose();
    }
});
// Default method for closing the frame
f.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});
// Adding the created objects

// to the frame
f.add(l1);
f.add(t1);
f.add(l2);
f.add(t2);
f.add(l3);
f.add(j1);
f.add(l4);
f.add(j2);
f.add(l5);
f.add(t3);
f.add(b1);
f.add(b2);
f.setLayout(null);
f.setSize(700, 600);
f.setVisible(true);
}
// Driver code
public static void main(String args[])
{
    StudentInfo();
}
}

```

Experiment 4

```
import java.sql.*;
public class InsertStaticOracle {
    public static void main(String[] args) {
        Connection con = null;
        Statement stmt = null;
        try {
            // Load Oracle JDBC Driver
            Class.forName("oracle.jdbc.driver.OracleDriver");

            // Connect to the database (corrected URL)
            con = DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe", // Corrected DB URL
                "system", // Replace with your DB username
                "123456789" // Replace with your DB password
            );
            // Create SQL statement for insertion
            stmt = con.createStatement();
            // Define the SQL INSERT statement
            String sql = "INSERT INTO student (ID, NAME) VALUES (101, 'xyz')";
            // Replace 'your_table_name' with the actual table name and columns with real column names
            // Execute the SQL statement
            stmt.executeUpdate(sql);
            System.out.println("Record inserted successfully.");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try { if (stmt != null) stmt.close(); } catch (Exception e) {}
            try { if (con != null) con.close(); } catch (Exception e) {}
        }
    }
}
```

//program to retrieve data

```
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
public class SelectOracle
{
    public static void main(String args[]) throws SQLException
    {
        System.out.println("Step 1: ");
        oracle.jdbc.driver.OracleDriver obj = new
        oracle.jdbc.driver.OracleDriver();
```

```

// Class.forName(oracle.jdbc.driver.OracleDriver);
System.out.println("Driver loaded successfully");

System.out.println("Step 2: ");
String url="jdbc:oracle:thin:@localhost:1521:XE",uname="system" ,
password="123456789";
Connection connection =
DriverManager.getConnection(url,uname,password);
if(connection!=null)
    System.out.println("Connection Established Successfully");
else
    System.out.println("Connection Not Established Successfully");
System.out.println("Step 3: ");
Statement st = connection.createStatement();
System.out.println("Statement Referenced ");
System.out.println("Step 4: ");
System.out.println("Step 5: ");
String qry = "select * from Student";
ResultSet rs = st.executeQuery(qry);
System.out.println("rs: "+rs);
System.out.println("Step 6: ");
System.out.println("Id\tName\taddress");
while(rs.next())
{
    int x = rs.getInt(1);
    String y = rs.getString("Name");String s = rs.getString(3);
    System.out.println(x+"\t"+y+"\t"+s);
}
System.out.println("Step 7: ");
rs.close();
st.close();

connection.close();
}

}

```

Experiment 5

```
import java.util.Scanner;
public class ChkPalindrome
{
    public static void main(String args[])
    {
        String str, rev = "";
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a string:");
        str = sc.nextLine();

        int length = str.length();

        for ( int i = length - 1; i >= 0; i-- )
            rev = rev + str.charAt(i);

        if (str.equals(rev))
            System.out.println(str+" is a palindrome");
        else
            System.out.println(str+" is not a palindrome");
    }
}
```


Experiment 6

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.Scanner;
public class RetriveIP
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Host Name: ");
        String host = sc.nextLine();
        try
        {
            InetAddress ip = InetAddress.getByName(host);
            System.out.println("IP Adress of Computer is:"+ip.getHostAddress());
        }
        catch(UnknownHostException e)
        {
            System.out.print(e);
        }
    }
}
```