1. Design your biodata by using various AWT components.

Ans:

a] Code:

```
import java.awt.*;

public class BiodataForm {
    public static void main(String[] args) {
        Frame frame = new Frame("Biodata Form");
        frame.setSize(500, 600);
        frame.setLayout(null);
        Label nameLabel = new Label("Name:");
        nameLabel.setBounds(50, 50, 100, 30);
        frame.add(nameLabel);
        Label ageLabel = new Label("Age:");
        ageLabel.setBounds(50, 100, 100, 30);
        frame.add(ageLabel);
        Label genderLabel = new Label("Gender:");
        genderLabel.setBounds(50, 150, 100, 30);
        frame.add(genderLabel);
        Label skillsLabel = new Label("Skills:");
        skillsLabel.setBounds(50, 200, 100, 30);
        frame.add(skillsLabel);
        Label countryLabel = new Label("Country:");
        countryLabel.setBounds(50, 250, 100, 30);
        frame.add(countryLabel);
        Label addressLabel = new Label("Address:");
        addressLabel.setBounds(50, 300, 100, 30);
        frame.add(addressLabel);
        TextField nameField = new TextField();
        nameField.setBounds(150, 50, 200, 30);
        frame.add(nameField);
        TextField ageField = new TextField();
        ageField.setBounds(150, 100, 200, 30);
        frame.add(ageField);
        CheckboxGroup genderGroup = new CheckboxGroup();
        Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
        maleCheckbox.setBounds(150, 150, 70, 30);
        frame.add(maleCheckbox);
        Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
        femaleCheckbox.setBounds(230, 150, 70, 30);
        frame.add(femaleCheckbox);
        Checkbox javaCheckbox = new Checkbox("Java");
        javaCheckbox.setBounds(150, 200, 70, 30);
        frame.add(javaCheckbox);
        Checkbox pythonCheckbox = new Checkbox("Python");
        pythonCheckbox.setBounds(230, 200, 70, 30);
        frame.add(pythonCheckbox);
        Checkbox cppCheckbox = new Checkbox("C++");
        cppCheckbox.setBounds(310, 200, 70, 30);
        frame.add(cppCheckbox);
        Choice countryChoice = new Choice();
        countryChoice.add("India");
        countryChoice.add("USA");
        countryChoice.add("UK");
        countryChoice.add("Canada");
        countryChoice.add("Australia");
        countryChoice.setBounds(150, 250, 200, 30);
        frame.add(countryChoice);
        TextArea addressArea = new TextArea();
        addressArea.setBounds(150, 300, 200, 100);
        frame.add(addressArea);
        Button submitButton = new Button("Submit");
        submitButton.setBounds(150, 420, 100, 30);
        frame.add(submitButton);
        frame.setVisible(true);
    }
}
```

2. Design an applet/Application using List components to add names of 10 different cities.

Ans:

a] Code:

```
/*<applet code=CityListApplet width=300 height=300></applet>*/
import java.awt.*;
import java.applet.*;
public class CityListApplet extends Applet {
  public void init() {
    setLayout(new FlowLayout());
    List cityList = new List(10, false);
    cityList.add("Mumbai");
    cityList.add("Kalyan");
    cityList.add("Thane");
    cityList.add("Delhi");
    cityList.add("Vadala");
    cityList.add("Ghatkopar");
    cityList.add("Kurla");
    cityList.add("Dadar");
    cityList.add("Cottongreen");
    cityList.add("Mumbai central");
    add(cityList);
  }
}
```

3. WAP to use Border Layout .

Ans:

a] Code:

```
/*
<applet code=BorderDemo width=300 height=300></applet>
*/
import java.awt.*;
import java.applet.*;
import java.util.*;

public class BorderDemo extends Applet {
public void init() {
setLayout(new BorderLayout());
add(new Button("This is across the top."),BorderLayout.NORTH);
add(new Label("The footer message might go here."),BorderLayout.SOUTH);
add(new Button("Right"), BorderLayout.EAST);
add(new Button("Left"), BorderLayout.WEST);
String msg = "Hello I am Siddharth Shah student of Vidyalankar Polytechnic\n" +
"I am in Computer Engineering branch " +
"I have scored 93% in sem 4.\n" +
" - Siddharth Paresh Shah\n\n";
add(new TextArea(msg), BorderLayout.CENTER);
}
}
```

4. WAP which creates Menu of different colors and disable menu item for Black color.

Ans:

a] Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class ColorMenuExample {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Color Menu Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 300);
    JMenuBar menuBar = new JMenuBar();
    JMenu colorMenu = new JMenu("Colors");
    JMenuItem redItem = new JMenuItem("Red");
    JMenuItem greenItem = new JMenuItem("Green");
    JMenuItem blueItem = new JMenuItem("Blue");
    JMenuItem blackItem = new JMenuItem("Black");
    redItem.addActionListener(e ->
frame.getContentPane().setBackground(Color.RED));
    greenItem.addActionListener(e ->
frame.getContentPane().setBackground(Color.GREEN));
    blueItem.addActionListener(e ->
frame.getContentPane().setBackground(Color.BLUE));
    blackItem.setEnabled(false);
    colorMenu.add(redItem);
    colorMenu.add(greenItem);
    colorMenu.add(blueItem);
    colorMenu.add(blackItem);
    menuBar.add(colorMenu);
    frame.setJMenuBar(menuBar);
    frame.getContentPane().setBackground(Color.WHITE);
    frame.setVisible(true);
  }
}
```

5. WAP to develop a frame to select the different states of India using JComboBox

Ans:

a] Code:

```
import javax.swing.*;
import java.awt.*;
public class JComboExample extends JFrame {
  JComboExample() {
    setTitle("State Selection");
    setSize(400, 200);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    Container contentPane = getContentPane();
    contentPane.setLayout(new FlowLayout());
    String[] states = { "Maharashtra", "Punjab", "Gujarat", "Madhya Pradesh",
"Haryana" };
    JComboBox<String> cb = new JComboBox<>(states);
    contentPane.add(cb);
  }
  public static void main(String[] args) {
    JFrame frame = new JComboExample();
    frame.setVisible(true);
  }
}
```

6. Develop a program to demonstrate the use of tree component in swing.

Ans:

a] Code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.tree.*;
/*
 <applet code="JTreeEvents" width=400 height=200>
 </applet>
*/
public class JTreeEvents extends JApplet {
 JTree tree;
 JTextField jtf;
 public void init() {
  Container contentPane = getContentPane();
  contentPane.setLayout(new BorderLayout());
  DefaultMutableTreeNode top = new DefaultMutableTreeNode("Options");
  DefaultMutableTreeNode a = new DefaultMutableTreeNode("CO");
  top.add(a);
  DefaultMutableTreeNode a1 = new DefaultMutableTreeNode("CO5I-A");
  a.add(a1);
  DefaultMutableTreeNode a2 = new DefaultMutableTreeNode("CO5I-B");
  a.add(a2);
  DefaultMutableTreeNode a3 = new DefaultMutableTreeNode("CO5I-C");
  a.add(a3);
  DefaultMutableTreeNode b = new DefaultMutableTreeNode("IF");
  top.add(b);
  DefaultMutableTreeNode b1 = new DefaultMutableTreeNode("IF5I-A");
  b.add(b1);
  DefaultMutableTreeNode b2 = new DefaultMutableTreeNode("IF5I-B");
  b.add(b2);
  DefaultMutableTreeNode b3 = new DefaultMutableTreeNode("IF5I-C");
  b.add(b3);
  tree = new JTree(top);
  int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
  int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
  JScrollPane jsp = new JScrollPane(tree, v, h);
  contentPane.add(jsp, BorderLayout.CENTER);
  jtf = new JTextField("", 20);
  contentPane.add(jtf, BorderLayout.SOUTH);
  tree.addMouseListener(new MouseAdapter() {
   public void mouseClicked(MouseEvent me) {
    doMouseClicked(me);
   }
  });
 }
 void doMouseClicked(MouseEvent me) {
  TreePath tp = tree.getPathForLocation(me.getX(), me.getY());
  if(tp != null)
   jtf.setText(tp.toString());
```

```
    else
      jtf.setText("");
  }
}
```

7. Develop a program to demonstrate the use of JTable.

Ans:

a] Code:

```
import java.awt.*;
import javax.swing.*;
/* <applet code="JTableDemo" width=400 height=200>  </applet>*/
public class JTableDemo extends JApplet {
  public void init() {
    Container contentPane = getContentPane();
    contentPane.setLayout(new BorderLayout());
    final String[] colHeads = { "Name", "Roll_no", "Percentage" };
    final Object[][] data = {
      { "Siddharth", "41", "97" },
      { "Aditya", "42", "97" },
      { "Darshan", "43", "93" },
      { "Sanjana", "44", "91" },
      { "Shrushti", "45", "88" },
      { "Shalaka", "46", "77" },
    };
    JTable table = new JTable(data, colHeads);
    int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
    int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
    JScrollPane jsp = new JScrollPane(table, v, h);
    contentPane.add(jsp, BorderLayout.CENTER);
  }
}
```

8. WAP to demonstrate various mouse events using MouseListener and MouseMotionListener interface

Ans:

a] Code:

```
import java.awt.*;

import java.applet.*;

import java.awt.event.*;

/*

<applet code="MouseEvents" width=300 height=300>

</applet>

*/

public class MouseEvents extends Applet implements MouseListener,
MouseMotionListener

{

  String msg = "";

  int mousex , mousey ; //

  public void init()

  {

    addMouseListener(this);

    addMouseMotionListener(this);

  }

  public void mouseClicked(MouseEvent me)

  {

    mousex = 0;

    mousey = 10;

    msg = "Mouse clicked.";
    repaint();
}
public void mouseEntered(MouseEvent me)
{
    mousex = 0;
    mousey = 10;
    msg = "Mouse entered.";
    repaint();
}
public void mouseExited(MouseEvent me)
{
    mousex = 0;
    mousey = 10;
    msg = "Mouse exited.";
    repaint();
}
public void mousePressed(MouseEvent me)
{
mousex = me.getX();
mousey = me.getY();
msg = "u pressed mouse";
repaint();
}
public void mouseReleased(MouseEvent me)
{
mousex = me.getX();
```

```
    mousey = me.getY();

    msg = "U released the mouse";

    repaint();

    }

    public void mouseDragged(MouseEvent me)

    {

    mousex = me.getX();

    mousey = me.getY();

    msg = "dragging mouse";

    repaint();

    }

    public void mouseMoved(MouseEvent me)

    {

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

    }

    public void paint(Graphics g)

    {

    g.drawString(msg, mousex, mousey);

    }

}
```

9. WAP to demonstrate the use of JTextfield and JPasswordField using Listener interface

Ans:

a] Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class UserAuth extends JFrame {
  private JTextField usernameField;
  private JPasswordField passwordField;
  public UserAuth() {
    setTitle("User Authentication");
    setLayout(new FlowLayout());
    add(new JLabel("Username:"));
    usernameField = new JTextField(20);
    add(usernameField);
    add(new JLabel("Password:"));
    passwordField = new JPasswordField(20);
    add(passwordField);
    JButton loginButton = new JButton("Login");
    add(loginButton);
    loginButton.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        authenticateUser();
      }
    });
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setSize(350, 150);
    setVisible(true);
  }
  private void authenticateUser() {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());
```

```
        if (username.equals("admin") && password.equals("password")) {
            JOptionPane.showMessageDialog(null, "Login successful!");
        } else {
            JOptionPane.showMessageDialog(null, "Invalid username or password");
        }
        usernameField.setText("");
        passwordField.setText("");
    }
    public static void main(String[] args) {
        new UserAuth();
    }
}
```

10. WAP to demonstrate the use of WindowAdapter class

Ans:

a] Code:

```
import java.awt.*;
import java.awt.event.*;
public class WindowAdapterDemo extends Frame {
    public WindowAdapterDemo() {
        setTitle("WindowAdapter Demo");
        setSize(400, 300);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.out.println("Window is closing");
                dispose();
            }
            public void windowOpened(WindowEvent we) {
                System.out.println("Window has opened");
            }
            public void windowIconified(WindowEvent we) {
                System.out.println("Window is minimized");
            }
            public void windowDeiconified(WindowEvent we) {
                System.out.println("Window is restored from minimized state");
            }
        });
    }
    public static void main(String[] args) {
        WindowAdapterDemo frame = new WindowAdapterDemo();
        frame.setVisible(true);
    }
}
```

11. WAP to demonstrate the use of InetAddress class and its factory methods

Ans:

a] Code:

```
import java.net.*;
class InetAddressTest {
  public static void main(String args[]) throws UnknownHostException {
    InetAddress Address = InetAddress.getLocalHost();
    System.out.println(Address);
    Address = InetAddress.getByName("google.com");
    System.out.println(Address);
    InetAddress SW[] = InetAddress.getAllByName("www.nba.com");
    for (int i = 0; i < SW.length; i++)
      System.out.println(SW[i]);
  }
}
```

12. WAP to demonstrate the use of URL and URLConnection class and its methods

Ans:

a] Code:

```
import java.io.*;
import java.net.*;
import java.util.Date;

public class UCDemo {
  public static void main(String[] args) throws Exception {
    int c;
    URL hp = new URL("http://www.internic.net");
    URLConnection hpcon = hp.openConnection();
    long d = hpcon.getDate();
    if (d == 0) {
      System.out.println("No date Information");
    } else {
      System.out.println("Date: " + new Date(d));
    }
    System.out.println("get-content type: " + hpcon.getContentType());
    d = hpcon.getExpiration();
    if (d == 0) {
      System.out.println("No Expiration Information");
    } else {
      System.out.println("Expire: " + new Date(d));
    }
    d = hpcon.getLastModified();
    if (d == 0) {
      System.out.println("No Last Modified Information");
    } else {
      System.out.println("Last Modified: " + new Date(d));
    }
    int len = hpcon.getContentLength();
    if (len == -1) {
      System.out.println("Content Length Unavailable");
    } else {
      System.out.println("Content Length: " + len);
    }
    if (len != 0) {
      System.out.println("===Content===");
      InputStream input = hpcon.getInputStream();
      int i = len;
      while ((c = input.read()) != -1) {
        System.out.print((char) c);
      }
      input.close();
    } else {
      System.out.println(" No Content Available");
    }
  }
}
```

13. WAP to insert and retrieve the data from database using JDBC

Ans:

a] Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package databasepractice;
import java.sql.*;

public class DATABASEPRACTICE {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
   static final String DB_URL = "jdbc:mysql://localhost/student";

   static final String USER = "root";
   static final String PASS = "";

    public static void main(String[] args) {
    Connection conn = null;
    Statement stmt = null;
   try{
      //STEP 2: Register JDBC driver
      Class.forName(JDBC_DRIVER);

      //STEP 3: Open a connection
      System.out.println("Connecting to a selected database...");
      conn = DriverManager.getConnection(DB_URL, USER, PASS);
      System.out.println("Connected database successfully...");

      //STEP 4: Execute a query
      System.out.println("Inserting records into the table...");
      stmt = conn.createStatement();
```

```
String sql = "INSERT INTO registration (name,percentage) VALUES ('ADITYA',99)";
stmt.executeUpdate(sql);
sql = "INSERT INTO registration (name,percentage) VALUES ('SIDDHARTH',99)";
stmt.executeUpdate(sql);
System.out.println("Inserted records into the table...");

String query = "SELECT * FROM registration";

// execute the query, and get a java resultset
ResultSet rs = stmt.executeQuery(query);

// iterate through the java resultset
while (rs.next())
{
  int stud_id = rs.getInt("stud_id");
  String name = rs.getString("name");
  int per = rs.getInt("percentage");

  // print the results
  System.out.format("%s, %s, %s\n", stud_id, name, per);
}
  }catch(SQLException se){
    //Handle errors for JDBC
    se.printStackTrace();
  }catch(Exception e){
    //Handle errors for Class.forName
    e.printStackTrace();
  }finally{
    //finally block used to close resources
    try{
      if(stmt!=null)
        conn.close();
    }catch(SQLException se){
      }// do nothing
      try{
         if(conn!=null)
            conn.close();
      }catch(SQLException se){
         se.printStackTrace();
      }//end finally try
   }//end try
   System.out.println("Goodbye!");

}
}
```

14. WAP servlet to send username and password using HTML forms and authenticate the user

Ans:

a] Code:

Servlet code-

```
import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import java.io.PrintWriter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
```

```
/**
 * LoginServlet handles user login authentication.
 */
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {

    /**
     * Handles HTTP POST requests for user authentication.
     *
     * @param request HttpServletRequest object
     * @param response HttpServletResponse object
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        try {
            if ("admin".equals(username) &&
"password123".equals(password)) {
```

```
                out.println("<h2>Login Successful!</h2>");
            } else {
                out.println("<h2>Invalid Username or Password.</h2>");
            }
        } finally {
            out.close(); // Ensure PrintWriter is closed to free resources
        }
    }

    /**
     * Provides a brief description of the servlet.
     *
     * @return A String describing the servlet
     */
    @Override
    public String getServletInfo() {
        return "LoginServlet: Handles user login authentication.";
    }
}
```

Html code-

```
     <!DOCTYPE html>
     <html>
     <head>
     <title>Login</title>
```

```
</head>
<body>
<h2>Login Form</h2>
<form action="LoginServlet" method="POST">
<label for="username">Username:</label>
<input type="text" id="username" name="username" required><br><br>
<label for="password">Password:</label>
<input type="password" id="password" name="password" required><br><br>
<button type="submit">Login</button>
</form>
</body>
</html>
```

Web.xml-

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
  <servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>LoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>

        <servlet-name>LoginServlet</servlet-name>

        <url-pattern>/LoginServlet</url-pattern>

    </servlet-mapping>

    <session-config>

        <session-timeout>

            30

        </session-timeout>

    </session-config>

</web-app>
```