1. Design your biodata by using various AWT components.

import java.awt.\*;

public class biodata extends Frame {

    public static void main(String[] args) {

        biodata f = new biodata();

        f.setSize(300, 200);

        f.setVisible(true);

        Label name = new Label("Name");

        Label nameval = new Label("Alpesh Bhagwatkar");

        Label roll = new Label("Roll No.");

        Label rollval = new Label("21203B0024");

        Label branch = new Label("Branch");

        Label branchval = new Label("CO");

        Label college = new Label("College");

        Label collegeval = new Label("VP");

        f.setLayout(new GridLayout(4, 2));

        f.add(name);

        f.add(nameval);

        f.add(roll);

        f.add(rollval);

        f.add(branch);

        f.add(branchval);

        f.add(college);

        f.add(collegeval);

    }

}

1. Design an applet/Application using listcomp to add names of 10 different cities.

import java.awt.\*;

import java.applet.\*;

/\* <applet code="listcomp" width="300" height="200"></applet> \*/

public class listcomp extends Applet {

    public void init() {

        List cities = new List(10);

        cities.add("Mumbai");

        cities.add("Pune");

        cities.add("Nagpur");

        cities.add("Nashik");

        cities.add("Aurangabad");

        cities.add("Kolhapur");

        cities.add("Solapur");

        cities.add("Amravati");

        cities.add("Akola");

        cities.add("Latur");

        add(cities);

        setSize(300, 200);

        setVisible(true);

    }

}

1. WAP to use Border Layout .

import java.awt.\*;

public class border extends Frame {

    public static void main(String[] args) {

        border f = new border();

        f.setSize(300, 200);

        f.setVisible(true);

        f.setLayout(new BorderLayout());

        Button east = new Button("East");

        Button west = new Button("West");

        Button north = new Button("North");

        Button south = new Button("South");

        f.add(east, BorderLayout.EAST);

        f.add(west, BorderLayout.WEST);

        f.add(north, BorderLayout.NORTH);

        f.add(south, BorderLayout.SOUTH);

    }

}

1. WAP which creates Menu of different colors and disable menuitem for Black color.

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

import java.awt.\*;

public class menuitem extends Frame {

    public static void main(String[] args) {

        menuitem f = new menuitem();

        f.setTitle("Menu");

        f.setSize(400, 400);

        f.setVisible(true);

        MenuBar mb = new MenuBar();

        f.setMenuBar(mb);

        Menu m1 = new Menu("Color");

        mb.add(m1);

        MenuItem mi1 = new MenuItem("Red");

        m1.add(mi1);

        MenuItem mi2 = new MenuItem("Green");

        m1.add(mi2);

        MenuItem mi3 = new MenuItem("Blue");

        m1.add(mi3);

        MenuItem mi4 = new MenuItem("Black");

        m1.add(mi4);

        mi4.setEnabled(false);

        MenuItem mi5 = new MenuItem("White");

        m1.add(mi5);

    }

}

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

import java.awt.\*;

 import javax.swing.\*;

public class combobox extends JFrame

 { combobox()

{ Container contentPane = getContentPane();

 setSize(500,500); setTitle("JComboBox");

 contentPane.setLayout(new FlowLayout());

 JComboBox states = new JComboBox();

states.addItem("Maharashtra");

states.addItem("Goa");

 states.addItem("Himachal Pradesh");

 contentPane.add(states);

setVisible(true);

 }

public static void main(String[] args)

{

combobox p = new combobox();

 } }

1. Develop a program to demonstrate the use of trees component in swing.

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

import java.awt.\*;

import javax.swing.\*;

import javax.swing.tree.DefaultMutableTreeNode;

public class trees extends JFrame {

    public static void main(String[] args) {

        trees f = new trees();

        f.setTitle("Tree");

        f.setSize(400, 400);

        f.setVisible(true);

        Container c = f.getContentPane();

        c.setLayout(new FlowLayout());

        DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");

        DefaultMutableTreeNode states = new DefaultMutableTreeNode("States");

        DefaultMutableTreeNode cars = new DefaultMutableTreeNode("Cars");

        DefaultMutableTreeNode andhra = new DefaultMutableTreeNode("Andhra Pradesh");

        DefaultMutableTreeNode arunachal = new DefaultMutableTreeNode("Arunachal Pradesh");

        DefaultMutableTreeNode assam = new DefaultMutableTreeNode("Assam");

        DefaultMutableTreeNode lamborghini = new DefaultMutableTreeNode("Lamborghini");

        DefaultMutableTreeNode ferrari = new DefaultMutableTreeNode("Ferrari");

        DefaultMutableTreeNode bugatti = new DefaultMutableTreeNode("Bugatti");

        root.add(states);

        root.add(cars);

        states.add(andhra);

        states.add(arunachal);

        states.add(assam);

        cars.add(lamborghini);

        cars.add(ferrari);

        cars.add(bugatti);

        JTree tree = new JTree(root);

        c.add(tree);

    }

}

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

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

import java.awt.\*;

import javax.swing.\*;

public class jtable extends JFrame {

    public static void main(String[] args) {

        jtable f = new jtable();

        f.setTitle("Table");

        f.setSize(400, 400);

        f.setVisible(true);

        Container c = f.getContentPane();

        c.setLayout(new FlowLayout());

        String data[][] = {{"1", "2", "3"}, {"4", "5", "6"}, {"7", "8", "9"}};

        String th[] = {"A", "B", "C"};

        JTable table = new JTable(data, th);

        c.add(table);

    }

}

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

// WAP to demonstrate various mouse events using MouseListener and MouseMotionListener interface

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class mouselistener extends JFrame implements MouseListener, MouseMotionListener {

    public static void main(String[] args) {

        mouselistener f = new mouselistener();

        f.setTitle("Mouse Events");

        f.setSize(400, 400);

        f.setVisible(true);

        Container c = f.getContentPane();

        c.setLayout(new FlowLayout());

        f.addMouseListener(f);

        f.addMouseMotionListener(f);

    }

    public void mouseClicked(MouseEvent e) {

        System.out.println("Mouse Clicked");

    }

    public void mouseEntered(MouseEvent e) {

        System.out.println("Mouse Entered");

    }

    public void mouseExited(MouseEvent e) {

        System.out.println("Mouse Exited");

    }

    public void mousePressed(MouseEvent e) {

        System.out.println("Mouse Pressed");

    }

    public void mouseReleased(MouseEvent e) {

        System.out.println("Mouse Released");

    }

    public void mouseDragged(MouseEvent e) {

        System.out.println("Mouse Dragged");

    }

    public void mouseMoved(MouseEvent e) {

        System.out.println("Mouse Moved");

    }

}

1. WAP to demonstrate the use of JTextfield and JPasswordField using Listener interface frame.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class frame extends JFrame implements ActionListener {

    private JTextField usernameField;

    private JPasswordField passwordField;

    public frame() {

        setLayout(new FlowLayout());

        usernameField = new JTextField(15);

        passwordField = new JPasswordField(15);

        JButton loginButton = new JButton("Login");

        loginButton.addActionListener(this);

        add(new JLabel("Username:"));

        add(usernameField);

        add(new JLabel("Password:"));

        add(passwordField);

        add(loginButton);

        setTitle("Login Demo");

        setSize(250, 150);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLocationRelativeTo(null);  // Center the frame

        setVisible(true);

    }

    public static void main(String[] args) {

        new frame();

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        String username = usernameField.getText();

        String password = new String(passwordField.getPassword());

        System.out.println("Username: " + username);

        System.out.println("Password: " + password);

    }

}

1. WAP to demonstrate the use of WindowAdapter class.

// WAP to demonstrate the use of WindowAdapter class

import java.awt.\*;

import java.awt.event.\*;

public class adapter extends Frame {

    public adapter() {

        setSize(400, 300);

        addWindowListener(new MyWindowAdapter());

        setVisible(true);

    }

    class MyWindowAdapter extends WindowAdapter {

        public void windowClosing(WindowEvent e) {

            System.out.println("Window is closing...");

            System.exit(0);

        }

        public void windowActivated(WindowEvent e) {

            System.out.println("Window is activated");

        }

        public void windowDeactivated(WindowEvent e) {

            System.out.println("Window is deactivated");

        }

    }

    public static void main(String[] args) {

        new adapter();

    }

}

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

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

import java.net.\*;

public class inetaddr {

    public static void main(String[] args) throws UnknownHostException{

        InetAddress localAddress = InetAddress.getLocalHost();

        System.out.println("Local Host Name: " + localAddress.getHostName());

        System.out.println("Local Host address: " + localAddress.getHostAddress());

        String website = "www.youtube.com";

        InetAddress address = InetAddress.getByName(website);

        System.out.println("Website Name: " + website);

        System.out.println("Host Name: " + address.getHostName());

        System.out.println("Host address: " + address.getHostAddress());

        InetAddress[] addresses = InetAddress.getAllByName(website);

        System.out.println("Website Name: " + website);

        for (int i = 0; i < addresses.length; i++){

            System.out.println("Host Name: " + addresses[i].getHostName());

            System.out.println("Host address: " + addresses[i].getHostAddress());

        }

    }

}

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

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

import java.net.\*;

public class urlconn {

    public static void main(String[] args) {

        try {

            URL url = new URL("https://www.google.com/");

            URLConnection urlConnection = url.openConnection();

            System.out.println("Protocol: " + url.getProtocol());

            System.out.println("Host Name: " + url.getHost());

            System.out.println("Port Number: " + url.getPort());

            System.out.println("Path: " + url.getPath());

            System.out.println("Content Type: " + urlConnection.getContentType());

            System.out.println("Content: " + urlConnection.getContent());

            System.out.println("Content Length: " + urlConnection.getContentLength());

            System.out.println("Date: " + urlConnection.getDate());

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}

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

// WAP to insert and retrieve the data from database using JDBC

import java.sql.\*;

public class q13 {

    public static void main(String[] args) throws Exception{

        String url = "jdbc:mysql://localhost:3306/";

        String uname = "root";

        String password = "durgesh";

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection con = DriverManager.getConnection(url, uname, password);

        // String query = "INSERT INTO student ('name', 'rollno', 'marks') VALUES ('Atharv', '1', '100');";

        // Statement st = con.createStatement();

        // int n = st.executeUpdate(query);

        // System.out.println(n + " row(s) affected");

        // query = "SELECT \* FROM student;";

        // ResultSet rs = st.executeQuery(query);

        // while(rs.next()){

        //     String userData = rs.getString("name") + " : " + rs.getString("rollno") + " : " + rs.getString("marks");

        //     System.out.println(userData);

        // }

        // st.close();

        // con.close();

    }

}

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

1. index.html

```html

<html>

<head>

<body>

<form method=get action="http://localhost:8080/atharv/servlet/password ">

Name:<input type="text" name="t1"><br>

password:<input type="password" name="t2"><br>

<input type="submit">

</form>

</body>

</html>

```

1. demo.java

```java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class demo extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws IOException, ServletException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("<html>");

out.println("<head>");

out.println("<title>Hello World!</title>");

out.println("</head>");

out.println("<body>");

String a = request.getParameter("t1");

String b = request.getParameter("t2");

if (b.length() <= 6) {

out.println("<h3> Welcome To Home Page<h3>");

} else {

out.println("<h3>Password Should not more than 6 Character<h3>");

}

out.println("</body>");

out.println("</html>");

}

}

```

1. web.xml

```xml

<web-app>

<servlet>

<servlet-name>password</servlet-name>

<servlet-class>password</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>password</servlet-name>

<url-pattern>/servlet/password</url-pattern>

</servlet-mapping>

</web-app>

```