1. Design your biodata by using various AWT components.

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

public class BiodataApp extends Frame {
    private Label nameLabel, ageLabel, genderLabel, addressLabel;
    private TextField nameTextField, ageTextField, addressTextField;
    private Choice genderChoice;
    private Button submitButton;

    public BiodataApp() {
        // Set layout manager
        setLayout(new GridLayout(5, 2));

        // Create components
        nameLabel = new Label("Name:");
        ageLabel = new Label("Age:");
        genderLabel = new Label("Gender:");
        addressLabel = new Label("Address:");

        nameTextField = new TextField();
        ageTextField = new TextField();
        genderChoice = new Choice();
        genderChoice.add("Male");
        genderChoice.add("Female");
        genderChoice.add("Other");

        addressTextField = new TextField();

        submitButton = new Button("Submit");

        // Add components to the frame
        add(nameLabel);
        add(nameTextField);
        add(ageLabel);
        add(ageTextField);
        add(genderLabel);
        add(genderChoice);
        add(addressLabel);
        add(addressTextField);
        add(submitButton);

        // Add event listener for the submit button
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Handle the submit button click event
```

```
                String name = nameTextField.getText();
                String age = ageTextField.getText();
                String gender = genderChoice.getSelectedItem();
                String address = addressTextField.getText();

                // Display the entered information
                System.out.println("Name: " + name);
                System.out.println("Age: " + age);
                System.out.println("Gender: " + gender);
                System.out.println("Address: " + address);
            }
        });

        // Set frame properties
        setTitle("Biodata Form");
        setSize(300, 200);
        setVisible(true);

        // Add window listener to handle closing event
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

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

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

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

public class CityListApp extends Frame {
    private List cityList;
    private TextField cityTextField;

    public CityListApp() {
        // Set layout manager
        setLayout(new FlowLayout());
```

```
        // Create List and TextField components
        cityList = new List(10);
        cityTextField = new TextField(15);

        // Create Add button
        Button addButton = new Button("Add");
        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addCity();
            }
        });

        // Add components to the frame
        add(new Label("Cities:"));
        add(cityList);
        add(new Label("Add City:"));
        add(cityTextField);
        add(addButton);

        // Set frame properties
        setTitle("City List App");
        setSize(300, 200);
        setVisible(true);

        // Add window listener to handle closing event
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    private void addCity() {
        String cityName = cityTextField.getText();
        if (!cityName.isEmpty()) {
            cityList.add(cityName);
            cityTextField.setText("");
        }
    }

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

3. WAP to use Border Layout .

```java
import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Frame;

public class BorderLayoutExample {

    public BorderLayoutExample() {
        // Create a frame
        Frame frame = new Frame("BorderLayout Example");

        // Create buttons
        Button buttonNorth = new Button("North");
        Button buttonSouth = new Button("South");
        Button buttonEast = new Button("East");
        Button buttonWest = new Button("West");
        Button buttonCenter = new Button("Center");

        // Set layout manager to BorderLayout
        frame.setLayout(new BorderLayout());

        // Add buttons to the frame with specified regions
        frame.add(buttonNorth, BorderLayout.NORTH);
        frame.add(buttonSouth, BorderLayout.SOUTH);
        frame.add(buttonEast, BorderLayout.EAST);
        frame.add(buttonWest, BorderLayout.WEST);
        frame.add(buttonCenter, BorderLayout.CENTER);

        // Set frame properties
        frame.setSize(400, 300);
        frame.setVisible(true);

        // Handle closing event
        frame.addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent
windowEvent) {
                System.exit(0);
            }
        });
    }

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

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

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

public class ColorMenuApp extends Frame {
    private CheckboxMenuItem blackMenuItem;

    public ColorMenuApp() {
        // Create a menu bar
        MenuBar menuBar = new MenuBar();

        // Create a menu
        Menu colorMenu = new Menu("Colors");

        // Create color menu items
        CheckboxMenuItem redMenuItem = new CheckboxMenuItem("Red");
        CheckboxMenuItem greenMenuItem = new CheckboxMenuItem("Green");
        CheckboxMenuItem blueMenuItem = new CheckboxMenuItem("Blue");
        blackMenuItem = new CheckboxMenuItem("Black");

        // Add action listeners to handle color selection
        redMenuItem.addItemListener(new ColorItemListener(Color.RED));
        greenMenuItem.addItemListener(new
ColorItemListener(Color.GREEN));
        blueMenuItem.addItemListener(new ColorItemListener(Color.BLUE));
        blackMenuItem.addItemListener(new
ColorItemListener(Color.BLACK));

        // Add color menu items to the menu
        colorMenu.add(redMenuItem);
        colorMenu.add(greenMenuItem);
        colorMenu.add(blueMenuItem);
        colorMenu.add(blackMenuItem);

        // Add the color menu to the menu bar
        menuBar.add(colorMenu);

        // Set the menu bar for the frame
        setMenuBar(menuBar);

        // Set frame properties
        setTitle("Color Menu App");
        setSize(300, 200);
        setVisible(true);

        // Handle closing event
```

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

    private class ColorItemListener implements ItemListener {
        private Color color;

        public ColorItemListener(Color color) {
            this.color = color;
        }

        public void itemStateChanged(ItemEvent e) {
            if (e.getStateChange() == ItemEvent.SELECTED) {
                // Handle color selection
                setBackground(color);

                // Disable the black menu item
                if (color.equals(Color.BLACK)) {
                    blackMenuItem.setEnabled(false);
                } else {
                    blackMenuItem.setEnabled(true);
                }
            }
        }
    }

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

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

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

public class StateSelectionApp {
    private JFrame frame;
    private JComboBox<String> stateComboBox;

    public StateSelectionApp() {
```

```
        // Create the frame
        frame = new JFrame("State Selection");

        // Create an array of Indian states
        String[] states = { "Goa", "Gujarat",
"Maharashtra","Manipur","Odisha", "Punjab", "Rajasthan", "Tamil Nadu"};

        // Create a JComboBox with the array of states
        stateComboBox = new JComboBox<>(states);

        // Create a button to display the selected state
        JButton showButton = new JButton("Show Selected State");
        showButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String selectedState = (String)
stateComboBox.getSelectedItem();
                JOptionPane.showMessageDialog(frame, "Selected State: " +
selectedState);
            }
        });

        // Set layout manager to default BorderLayout
        frame.setLayout(new java.awt.BorderLayout());

        // Add the JComboBox to the frame's content pane
        frame.add(stateComboBox, java.awt.BorderLayout.NORTH);

        // Add the button to display the selected state
        frame.add(showButton, java.awt.BorderLayout.SOUTH);

        // Set frame properties
        frame.setSize(300, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        // Create the StateSelectionApp object
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new StateSelectionApp();
            }
        });
    }
}
```

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

```
import javax.swing.*;
import javax.swing.tree.DefaultMutableTreeNode;

public class TreeDemoApp {
    private JFrame frame;

    public TreeDemoApp() {
        // Create the frame
        frame = new JFrame("Tree Demo");

        // Create a root node for the tree
        DefaultMutableTreeNode rootNode = new
DefaultMutableTreeNode("Categories");

        // Create child nodes
        DefaultMutableTreeNode fruitsNode = new
DefaultMutableTreeNode("Fruits");
        fruitsNode.add(new DefaultMutableTreeNode("Apple"));
        fruitsNode.add(new DefaultMutableTreeNode("Banana"));
        fruitsNode.add(new DefaultMutableTreeNode("Orange"));

        DefaultMutableTreeNode vegetablesNode = new
DefaultMutableTreeNode("Vegetables");
        vegetablesNode.add(new DefaultMutableTreeNode("Carrot"));
        vegetablesNode.add(new DefaultMutableTreeNode("Broccoli"));
        vegetablesNode.add(new DefaultMutableTreeNode("Spinach"));

        // Add child nodes to the root node
        rootNode.add(fruitsNode);
        rootNode.add(vegetablesNode);

        // Create a JTree with the root node
        JTree tree = new JTree(rootNode);

        // Set layout manager to default BorderLayout
        frame.setLayout(new java.awt.BorderLayout());

        // Add the JTree to the frame's content pane
        frame.add(new JScrollPane(tree), java.awt.BorderLayout.CENTER);

        // Set frame properties
        frame.setSize(300, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
```

```
    public static void main(String[] args) {
        // Create the TreeDemoApp object
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new TreeDemoApp();
            }
        });
    }
}
```

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

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class StudentTableDemo {
    private JFrame frame;
    private JTable studentTable;

    public StudentTableDemo() {
        // Create the frame
        frame = new JFrame("Student Table Demo");

        // Create column names
        String[] columnNames = {"Roll Number", "Name", "Age", "Grade"};

        // Create data for the table
        Object[][] data = {
                {46 , "Jidnesh chavan", 19 , "A"},
                {41 , "Chirag sharma", 19, "A"},
                {43 , "Ashmeet bhatt ", 19, "A"},
        };

        // Create a DefaultTableModel with data and column names
        DefaultTableModel model = new DefaultTableModel(data,
columnNames);

        // Create a JTable with the DefaultTableModel
        studentTable = new JTable(model);

        // Set layout manager to default BorderLayout
        frame.setLayout(new java.awt.BorderLayout());

        // Add the JTable to the frame's content pane inside a JScrollPane
        frame.add(new JScrollPane(studentTable),
java.awt.BorderLayout.CENTER);
```

```
        // Set frame properties
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        // Create the StudentTableDemo object
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new StudentTableDemo();
            }
        });
    }
}
```

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

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

public class MouseEventDemoApp extends JFrame implements MouseListener,
MouseMotionListener {
    private JLabel statusLabel;

    public MouseEventDemoApp() {
        setTitle("Mouse Event Demo");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create a label to display mouse event information
        statusLabel = new JLabel("No Mouse Event");

        // Add mouse listeners to the frame
        addMouseListener(this);
        addMouseMotionListener(this);

        // Set layout manager to default BorderLayout
        setLayout(new BorderLayout());

        // Add the label to the frame's content pane
        add(statusLabel, BorderLayout.SOUTH);
```

```
        setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new MouseEventDemoApp();
            }
        });
    }

    // MouseListener methods
    public void mouseClicked(MouseEvent e) {
        statusLabel.setText("Mouse Clicked at (" + e.getX() + ", " +
e.getY() + ")");
    }

    public void mousePressed(MouseEvent e) {
        statusLabel.setText("Mouse Pressed at (" + e.getX() + ", " +
e.getY() + ")");
    }

    public void mouseReleased(MouseEvent e) {
        statusLabel.setText("Mouse Released at (" + e.getX() + ", " +
e.getY() + ")");
    }

    public void mouseEntered(MouseEvent e) {
        statusLabel.setText("Mouse Entered at (" + e.getX() + ", " +
e.getY() + ")");
    }

    public void mouseExited(MouseEvent e) {
        statusLabel.setText("Mouse Exited at (" + e.getX() + ", " + e.getY()
+ ")");
    }

    // MouseMotionListener methods
    public void mouseMoved(MouseEvent e) {
        statusLabel.setText("Mouse Moved at (" + e.getX() + ", " + e.getY()
+ ")");
    }

    public void mouseDragged(MouseEvent e) {
        statusLabel.setText("Mouse Dragged at (" + e.getX() + ", " +
e.getY() + ")");
    }
}
```

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

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

public class TextFieldPasswordDemoApp extends JFrame implements
ActionListener {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;

    public TextFieldPasswordDemoApp() {
        setTitle("Login Demo");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create JTextField and JPasswordField
        usernameField = new JTextField(15);
        passwordField = new JPasswordField(15);

        // Create JButton
        loginButton = new JButton("Login");

        // Add ActionListener to the button
        loginButton.addActionListener(this);

        // Set layout manager to default FlowLayout
        setLayout(new java.awt.FlowLayout());

        // Add components to the frame's content pane
        add(new JLabel("Username:"));
        add(usernameField);
        add(new JLabel("Password:"));
        add(passwordField);
        add(loginButton);

        setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
```

```
                new TextFieldPasswordDemoApp();
            }
        });
    }

    // ActionListener method
    public void actionPerformed(ActionEvent e) {
        // Get the text from the JTextField and JPasswordField
        String username = usernameField.getText();
        char[] passwordChars = passwordField.getPassword();
        String password = new String(passwordChars);

        // Check if the username and password are not empty
        if (!username.isEmpty() && !password.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Login
successful!\nUsername: " + username + "\nPassword: " + password);
        } else {
            JOptionPane.showMessageDialog(this, "Please enter both
username and password.", "Error", JOptionPane.ERROR_MESSAGE);
        }

        // Clear the password field after checking
        passwordField.setText("");
    }
}
```

10. WAP to demonstrate the use of WindowAdapter class

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

public class WindowAdapterDemoApp extends JFrame {
    public WindowAdapterDemoApp() {
        setTitle("WindowAdapter Demo");
        setSize(300, 200);

        // Add a WindowAdapter to handle window events
        addWindowListener(new WindowAdapter() {
            @Override
            public void windowOpened(WindowEvent e) {
                // Executed when the window is first opened
                System.out.println("Window opened!");
            }

            @Override
```

```
            public void windowClosing(WindowEvent e) {
                // Executed when the user clicks the close button
                System.out.println("Window closing...");
                dispose();  // Close the window
            }

            @Override
            public void windowClosed(WindowEvent e) {
                // Executed after the window has been closed
                System.out.println("Window closed!");
            }
        });

        setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new WindowAdapterDemoApp());
    }
}
```

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

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class InetAddressDemo {
    public static void main(String[] args) {
        try {
            // Using factory methods to get InetAddress instances
            InetAddress localHost = InetAddress.getLocalHost();
            InetAddress googleAddress =
InetAddress.getByName("www.google.com");
            InetAddress[] allGoogleAddresses =
InetAddress.getAllByName("www.google.com");

            // Display information about the local host
            System.out.println("Local Host:");
            System.out.println("Host Name: " + localHost.getHostName());
            System.out.println("Host Address: " +
localHost.getHostAddress());
            System.out.println();

            // Display information about Google's address
            System.out.println("Google's Address:");
```

```
            System.out.println("Host Name: " +
googleAddress.getHostName());
            System.out.println("Host Address: " +
googleAddress.getHostAddress());
            System.out.println();

            // Display information about all Google's addresses
            System.out.println("All Google Addresses:");
            for (InetAddress address : allGoogleAddresses) {
                System.out.println("Host Name: " +
address.getHostName());
                System.out.println("Host Address: " +
address.getHostAddress());
                System.out.println();
            }
        } catch (UnknownHostException e) {
            e.printStackTrace();
        }
    }
}
```

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

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;

public class URLConnectionDemo {
    public static void main(String[] args) {
        try {
            // Create a URL object
            URL url = new URL("https://www.example.com");

            // Open a connection to the URL
            URLConnection connection = url.openConnection();

            // Display information about the URL
            System.out.println("URL Information:");
            System.out.println("Protocol: " + url.getProtocol());
            System.out.println("Host: " + url.getHost());
            System.out.println("Port: " + url.getPort());
            System.out.println("Path: " + url.getPath());
            System.out.println("Query: " + url.getQuery());
```

```
            System.out.println();

            // Display information about the URLConnection
            System.out.println("URLConnection Information:");
            System.out.println("Content Type: " +
connection.getContentType());
            System.out.println("Content Length: " +
connection.getContentLength());
            System.out.println("Last Modified: " +
connection.getLastModified());
            System.out.println();

            // Read content from the URL
            BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            System.out.println("Content from the URL:");
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }

            // Close the BufferedReader
            reader.close();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

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

```
import java.sql.*;

public class JDBCDemo {
    // JDBC URL, username, and password of MySQL server
    private static final String JDBC_URL =
"jdbc:mysql://localhost:3306/your_database";
    private static final String USERNAME = "your_username";
    private static final String PASSWORD = "your_password";

    public static void main(String[] args) {
        try {
            // Load the JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish a connection
```

```
            try (Connection connection =
DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD)) {

                // Insert Data
                insertData(connection, "John Doe", 25);

                // Retrieve Data
                retrieveData(connection);

            } catch (SQLException e) {
                e.printStackTrace();
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    // Insert data into the database
    private static void insertData(Connection connection, String name, int
age) throws SQLException {
        String insertQuery = "INSERT INTO users (name, age) VALUES (?, ?)";

        try (PreparedStatement preparedStatement =
connection.prepareStatement(insertQuery)) {
            preparedStatement.setString(1, name);
            preparedStatement.setInt(2, age);

            int rowsAffected = preparedStatement.executeUpdate();
            System.out.println(rowsAffected + " row(s) affected by
insertion.");
        }
    }

    // Retrieve data from the database
    private static void retrieveData(Connection connection) throws
SQLException {
        String selectQuery = "SELECT * FROM users";

        try (Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery(selectQuery)) {

            System.out.println("\nRetrieved Data:");
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                int age = resultSet.getInt("age");
```

```
                System.out.println("ID: " + id + ", Name: " + name + ", Age:
" + age);
            }
        }
    }
}
```

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