Java Swing — Introduction

Swing is a part of Java Foundation Classes (JFC).
It is used to create Graphical User Interface (GUI) applications in Java.

- Lightweight components.
- Pluggable Look and Feel
- Platform-independent.
- More powerful than the old AWT (Abstract Window Toolkit).

Basic Concepts

- JFrame → Main window (like a blank page).
- JButton → A clickable button.
- JLabel → Display text.
- JTextField → Single-line text box.
- JTextArea → Multi-line text area.

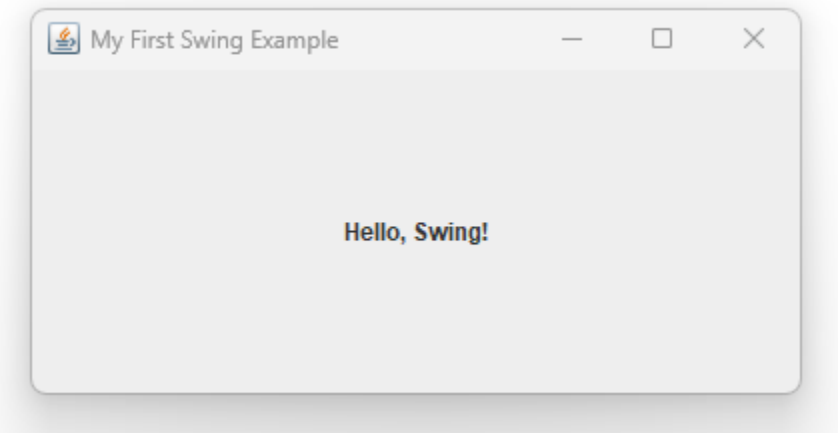
First Simple Example: "Hello Swing"

HelloSwing.java

```java
import javax.swing.*;  // import Swing classes
public class HelloSwing {
    public static void main(String[] args) {
        // Create a new frame (window)
        JFrame frame = new JFrame("My First Swing Example");
        // Create a label
        JLabel label = new JLabel("Hello, Swing!", JLabel.CENTER);
        // Set size of the frame
        frame.setSize(400, 200);
        // Add label to the frame
        frame.add(label);
        // Set default close operation
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Make the frame visible
        frame.setVisible(true);
    }
}
```

Output:

```
D:\Java Code\swing>javac HelloSwing.java

D:\Java Code\swing>java HelloSwing
```



Explanation

- JFrame creates a window.
- setSize(width, height) sets the window size.
- add(component) adds a label (or button) inside the frame.
- setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) ensures the program closes when you click X.
- setVisible(true) shows the window.

Adding Buttons and Action Listeners in Swing

In Swing, when you click a button, you can make something happen, like showing a message!

Important things to use:

- JButton → To create a button.
- ActionListener → To listen when the button is clicked.
- addActionListener() → To connect the button to an action.

setBounds() method

In Java Swing, the setBounds() method is used to set the position and size of a component (like a button, label, text field, etc.) on a container (like a JFrame or JPanel).

component.setBounds(int x, int y, int width, int height);

Parameters it takes:

- x – the x-coordinate (horizontal position) of the component's top-left corner.
- y – the y-coordinate (vertical position) of the component's top-left corner.
- width – the width of the component.
- height – the height of the component.

Example:

```
JButton button = new JButton("Click Me");

button.setBounds(100, 50, 120, 30);

frame.add(button);
```

- Here, the button will be placed 100 pixels from the left edge and 50 pixels from the top edge of the container.
- The button's size will be 120 pixels wide and 30 pixels tall.

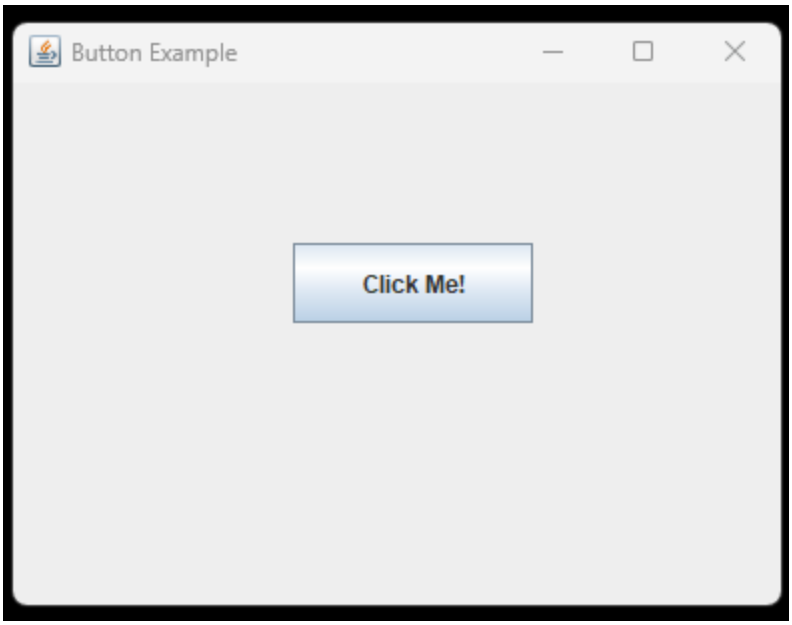
Example: Button with ActionListener

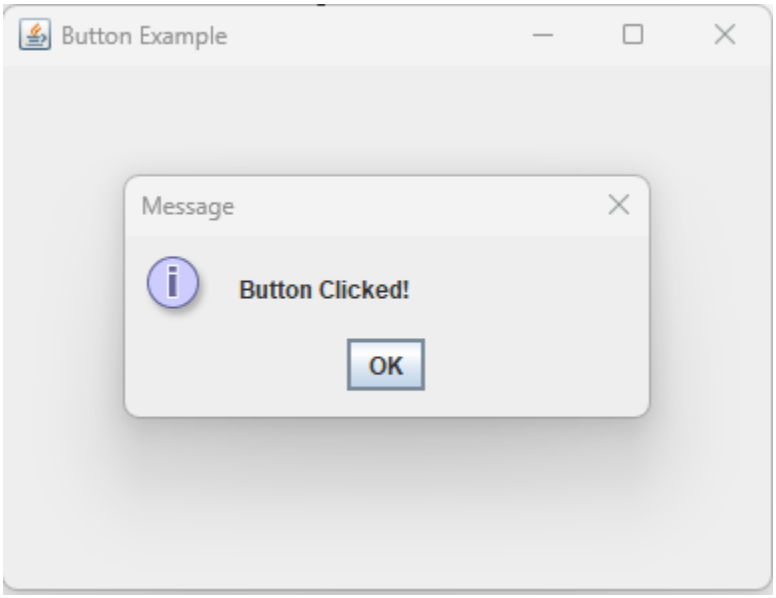
ButtonExample.java

```java
import javax.swing.*;
import java.awt.event.*;  // Needed for ActionListener
public class ButtonExample {
    public static void main(String[] args) {
        // Create a new frame
        JFrame frame = new JFrame("Button Example");
        // Create a new button
        JButton button = new JButton("Click Me!");
        // Set button position and size
        button.setBounds(140, 80, 120, 40);
        // Add ActionListener to the button
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Action performed when button is clicked
                JOptionPane.showMessageDialog(frame, "Button Clicked!");
            } });
        // Add button to the frame
        frame.add(button);
        // Set frame size
        frame.setSize(400, 300);
        // Set layout to null (absolute positioning)
        frame.setLayout(null);
        // Set default close operation
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Make frame visible
        frame.setVisible(true);
    }}
```

Output:

```
D:\Java Code\swing>javac ButtonExample.java

D:\Java Code\swing>java ButtonExample
```





Explanation

Thing	**Purpose**
JButton	Creates the clickable button
setBounds(x, y, width, height)	Sets position and size manually

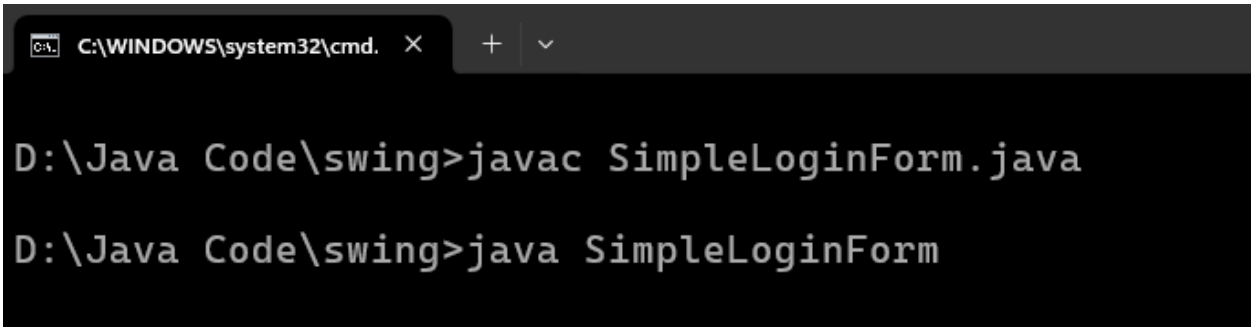
Thing	Purpose
addActionListener()	Listens for clicks
JOptionPane.showMessageDialog()	Shows a small message box

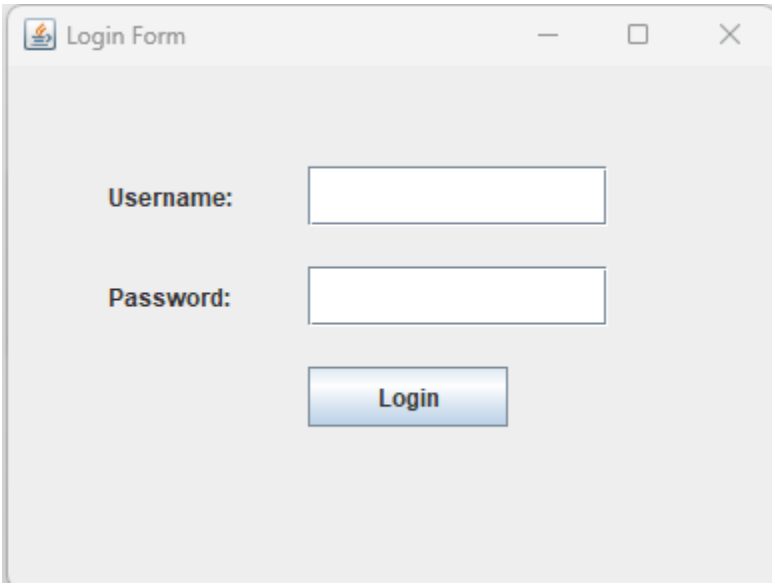
Mini Project: Simple Login Form (Swing + ActionListener)

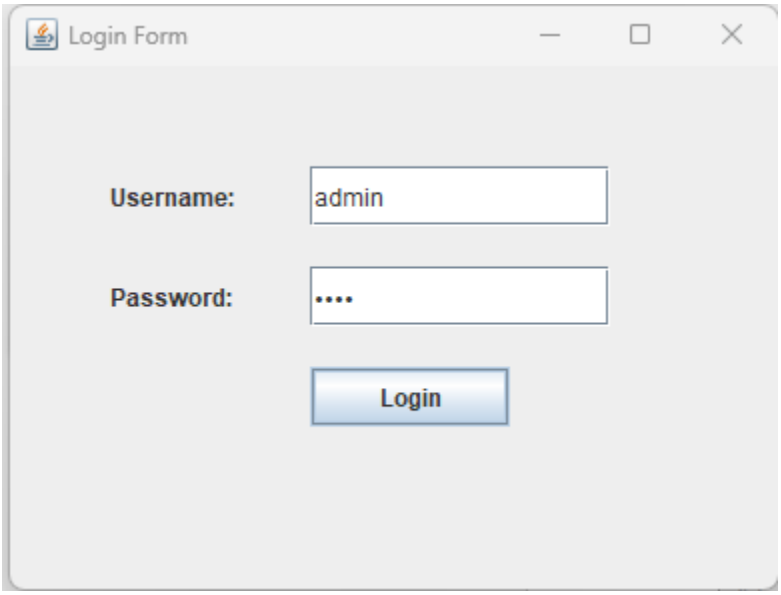
SimpleLoginForm.java

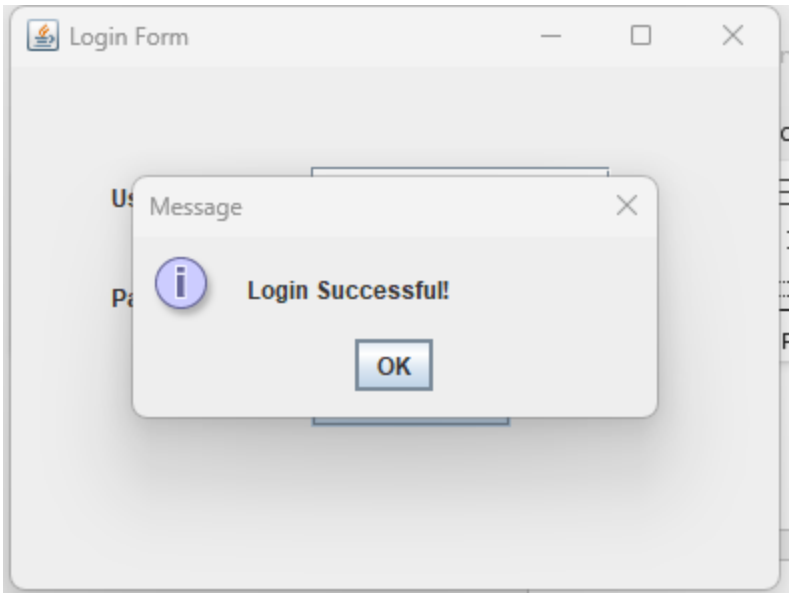
```java
import javax.swing.*;
import java.awt.event.*;  // For ActionListener
public class SimpleLoginForm {
    public static void main(String[] args) {
        // Create a frame
        JFrame frame = new JFrame("Login Form");
        // Create labels
        JLabel userLabel = new JLabel("Username:");
        JLabel passLabel = new JLabel("Password:");
        // Create text fields
        JTextField userText = new JTextField();
        JPasswordField passText = new JPasswordField();
        // Create button
        JButton loginButton = new JButton("Login");
        // Set positions and sizes (x, y, width, height)
        userLabel.setBounds(50, 50, 100, 30);
        passLabel.setBounds(50, 100, 100, 30);
        userText.setBounds(150, 50, 150, 30);
        passText.setBounds(150, 100, 150, 30);
        loginButton.setBounds(150, 150, 100, 30);
        // Add action listener to the button
        loginButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String username = userText.getText();
                String password = new String(passText.getPassword());

                if (username.equals("admin") && password.equals("1234")) {
                    JOptionPane.showMessageDialog(frame, "Login Successful!");
                } else {
                    JOptionPane.showMessageDialog(frame, "Invalid username or password");
                }
            }
        });
        // Add components to the frame
        frame.add(userLabel);
        frame.add(passLabel);
        frame.add(userText);
        frame.add(passText);
        frame.add(loginButton);
        // Frame settings
        frame.setSize(400, 300);
        frame.setLayout(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

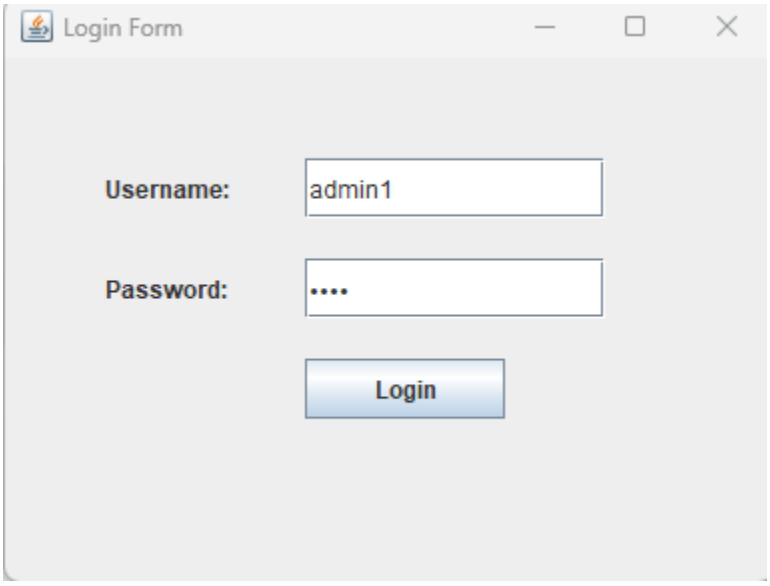
Output:

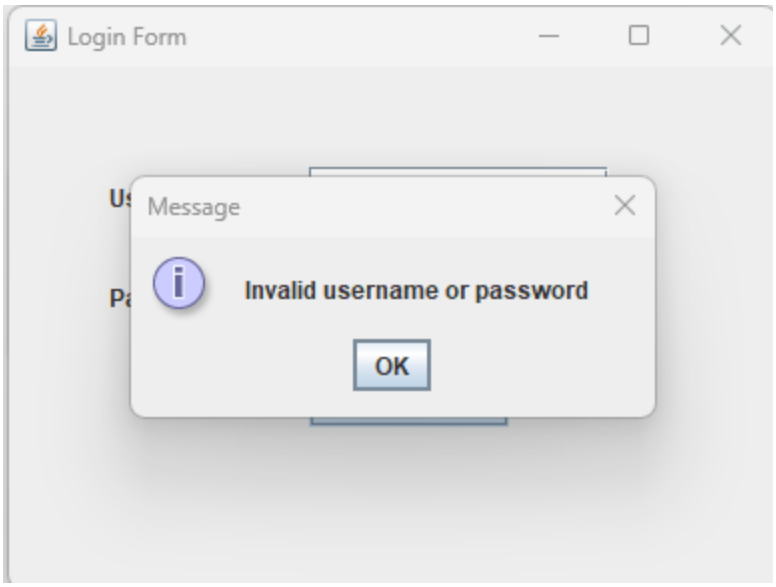

C:\WINDOWS\system32\cmd.
D:\Java Code\swing>javac SimpleLoginForm.java
D:\Java Code\swing>java SimpleLoginForm


Login Form
Username:
Password:
Login


Login Form
Username:
admin
Password:
Login


Login Form
Message
Login Successful!
OK


Login Form
Username:
admin1
Password:
Login



Explanation

Component	Purpose
JLabel	To display "Username" and "Password" labels
JTextField	To type username
JPasswordField	To type password
JButton	To click and check login
addActionListener()	To handle the button click

JDBC Example:

Here's the Java program to connect to the database, insert employee records, and display them.

```
import java.sql.*;  // For JDBC classes
import javax.swing.*;
public class SimpleJDBCExample {
    public static void main(String[] args) {
        // Database connection info
        String url = "jdbc:mysql://localhost:3306/company"; // Your MySQL database URL
        String username = "root";  // Your database username
        String password = "password";  // Your database password
        // JDBC Connection & Statement objects
        Connection conn = null;
        Statement stmt = null;
        try {
            // Connect to the database
            conn = DriverManager.getConnection(url, username, password);
            stmt = conn.createStatement();
            // SQL query to insert a new employee
            String insertSQL = "INSERT INTO emp (name, salary) VALUES ('Aman', 50000.00)";
            stmt.executeUpdate(insertSQL);
            // SQL query to select all employees from 'emp' table
            String selectSQL = "SELECT * FROM emp";
            ResultSet rs = stmt.executeQuery(selectSQL);
            // Display the result set (emp details)
            while (rs.next()) {
                int empId = rs.getInt("emp_id");
```

```
            String name = rs.getString("name");
            double salary = rs.getDouble("salary");
            // Show employee data in a message dialog
            JOptionPane.showMessageDialog(null, "Emp ID: " + empId + "\nName: " + name +
"\nSalary: " + salary);
        }
        // Close the ResultSet and Statement
        rs.close();
        stmt.close();
      } catch (SQLException e) {
        // Handle SQL exceptions
        e.printStackTrace();
      } finally {
        try {
          // Close the connection
          if (conn != null) {
            conn.close();
          }
        } catch (SQLException e) {
          e.printStackTrace();
        }
      }
  }
}
```

Java Swing Event Handling - Step-by-Step Programs

Step 1: Basic Frame with Button

```
import javax.swing.*;
public class Step1_BasicFrame {
public static void main(String[] args) {
JFrame frame = new JFrame("Step 1: Basic Frame");
JButton button = new JButton("Click Me");
button.setBounds(100, 100, 120, 40);
frame.add(button);
frame.setSize(300, 300);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Step 2: Button Click Event Handling

```
import javax.swing.*;
import java.awt.event.*;
public class Step2_ButtonClickEvent {
public static void main(String[] args) {
JFrame frame = new JFrame("Step 2: Button Event");
JButton button = new JButton("Click Me");
button.setBounds(100, 100, 120, 40);
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(frame, "Button Clicked!");
}
});
frame.add(button);
frame.setSize(300, 300);
frame.setLayout(null);
```

```
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Step 3: Text Input Event Handling

```
import javax.swing.*;
import java.awt.event.*;
public class Step3_TextFieldEvent {
public static void main(String[] args) {
JFrame frame = new JFrame("Step 3: Text Input");
JTextField textField = new JTextField();
JButton button = new JButton("Submit");
textField.setBounds(50, 50, 200, 30);
button.setBounds(100, 100, 100, 30);
button.addActionListener(e -> {
String name = textField.getText();
JOptionPane.showMessageDialog(frame, "Hello, " + name);
});
frame.add(textField);
frame.add(button);
frame.setSize(300, 250);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Step 4: CheckBox and RadioButton

```
import javax.swing.*;
import java.awt.event.*;
public class Step4_CheckBoxRadioButton {
public static void main(String[] args) {
```

```
JFrame frame = new JFrame("Step 4: CheckBox & Radio");
JCheckBox cbJava = new JCheckBox("Java");
JCheckBox cbPython = new JCheckBox("Python");
JRadioButton rMale = new JRadioButton("Male");
JRadioButton rFemale = new JRadioButton("Female");
ButtonGroup genderGroup = new ButtonGroup();
genderGroup.add(rMale);
genderGroup.add(rFemale);
JButton button = new JButton("Submit");
cbJava.setBounds(50, 30, 100, 30);
cbPython.setBounds(150, 30, 100, 30);
rMale.setBounds(50, 70, 100, 30);
rFemale.setBounds(150, 70, 100, 30);
button.setBounds(100, 120, 100, 30);
button.addActionListener(e -> {
String skills = "Skills: ";
if (cbJava.isSelected()) skills += "Java ";
if (cbPython.isSelected()) skills += "Python ";
String gender = rMale.isSelected() ? "Male" : (rFemale.isSelected() ?
"Female" : "Not selected");
JOptionPane.showMessageDialog(frame, skills + "\nGender: " + gender);
});
frame.add(cbJava); frame.add(cbPython);
frame.add(rMale); frame.add(rFemale);
frame.add(button);
frame.setSize(350, 230);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Step 5: ComboBox and JList

```
import javax.swing.*;
import java.awt.event.*;
public class Step5_ComboBoxList {
public static void main(String[] args) {
JFrame frame = new JFrame("Step 5: ComboBox & List");
String[] countries = {"India", "USA", "UK", "Germany"};
String[] colors = {"Red", "Green", "Blue", "Yellow"};
JComboBox<String> countryBox = new JComboBox<>(countries);
JList<String> colorList = new JList<>(colors);
JButton button = new JButton("Show Selection");
countryBox.setBounds(50, 30, 150, 30);
colorList.setBounds(50, 70, 100, 80);
button.setBounds(50, 170, 150, 30);
button.addActionListener(e -> {
String country = (String) countryBox.getSelectedItem();
String color = colorList.getSelectedValue();
JOptionPane.showMessageDialog(frame, "Country: " + country + "\nColor: " +
color);
});
frame.add(countryBox);
frame.add(colorList);
frame.add(button);
frame.setSize(300, 300);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Step 6: Mouse Events

```
import javax.swing.*;
```

```
import java.awt.event.*;
public class Step6_MouseEvents {
public static void main(String[] args) {
JFrame frame = new JFrame("Step 6: Mouse Events");
JLabel label = new JLabel("Click anywhere!");
label.setBounds(80, 50, 200, 30);
frame.addMouseListener(new MouseAdapter() {
public void mouseClicked(MouseEvent e) {
label.setText("Clicked at: " + e.getX() + ", " + e.getY());
}
});
frame.add(label);
frame.setSize(300, 200);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Step 7: Window Events

```
import javax.swing.*;
import java.awt.event.*;
public class Step7_WindowEvents {
public static void main(String[] args) {
JFrame frame = new JFrame("Step 7: Window Events");
JLabel label = new JLabel("Try minimizing or closing this window.");
label.setBounds(30, 50, 300, 30);
frame.addWindowListener(new WindowAdapter() {
public void windowIconified(WindowEvent e) {
System.out.println("Window Minimized");
}
public void windowDeiconified(WindowEvent e) {
```

```
System.out.println("Window Restored");
}
public void windowClosing(WindowEvent e) {
System.out.println("Window is closing...");
JOptionPane.showMessageDialog(frame, "Closing App...");
System.exit(0);
}
});
frame.add(label);
frame.setSize(350, 200);
frame.setLayout(null);
frame.setVisible(true);
}
}
```

Mini-Project: Swing-Based Student Feedback Form

Description:

This project demonstrates a Swing-based GUI for student feedback collection. It integrates JTextField,
JRadioButton, JComboBox, JList, JCheckBox, and event handling for buttons to dynamically display the
user's input.

Complete Code:

```
import javax.swing.*;
import java.awt.event.*;
public class StudentFeedbackForm {
public static void main(String[] args) {
JFrame frame = new JFrame("Student Feedback Form");
JLabel nameLabel = new JLabel("Name:");
JTextField nameField = new JTextField();
JLabel genderLabel = new JLabel("Gender:");
JRadioButton male = new JRadioButton("Male");
```

```
JRadioButton female = new JRadioButton("Female");
ButtonGroup genderGroup = new ButtonGroup();
JLabel courseLabel = new JLabel("Course:");
String[] courses = {"Java", "Python", "C++"};
JComboBox<String> courseBox = new JComboBox<>(courses);
JLabel ratingLabel = new JLabel("Rate the course:");
JList<String> ratingList = new JList<>(new String[]{"Excellent", "Good", "Average",
"Poor"});
JCheckBox practical = new JCheckBox("Practical");
JCheckBox theory = new JCheckBox("Theory");
JButton submitButton = new JButton("Submit");
JLabel outputLabel = new JLabel("");
// Setting Bounds
nameLabel.setBounds(30, 20, 100, 25);
nameField.setBounds(140, 20, 150, 25);
genderLabel.setBounds(30, 60, 100, 25);
male.setBounds(140, 60, 70, 25);
female.setBounds(210, 60, 80, 25);
courseLabel.setBounds(30, 100, 100, 25);
courseBox.setBounds(140, 100, 150, 25);
ratingLabel.setBounds(30, 140, 120, 25);
ratingList.setBounds(140, 140, 150, 60);
practical.setBounds(30, 210, 100, 25);
theory.setBounds(140, 210, 100, 25);
submitButton.setBounds(100, 250, 120, 30);
outputLabel.setBounds(30, 290, 300, 40);
// Adding components
genderGroup.add(male);
genderGroup.add(female);
frame.add(nameLabel);
frame.add(nameField);
```

```
frame.add(genderLabel);
frame.add(male);
frame.add(female);
frame.add(courseLabel);
frame.add(courseBox);
frame.add(ratingLabel);
frame.add(ratingList);
frame.add(practical);
frame.add(theory);
frame.add(submitButton);
frame.add(outputLabel);
// Action Listener
submitButton.addActionListener(e -> {
String name = nameField.getText();
String gender = male.isSelected() ? "Male" : (female.isSelected() ? "Female" :
"Unspecified");
String course = (String) courseBox.getSelectedItem();
String rating = ratingList.getSelectedValue();
String modules = "";
if (practical.isSelected()) modules += "Practical ";
if (theory.isSelected()) modules += "Theory";
outputLabel.setText("<html>Name: " + name + "<br>Gender: " + gender +
"<br>Course: " + course + "<br>Rating: " + rating +
"<br>Modules: " + modules + "</html>");
});
frame.setSize(350, 400);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```