**National University of Computer & Emerging Sciences, Karachi**

**Computer Science Department**

**Fall 2024, Lab Manual - 03**

|  |  |
| --- | --- |
| **Course Code: SL3001** | **Course: Software Development and construction** |
| **Instructor:** | **Yasir Arfat.** |

**Lab # 03**

**SWING API:**

It contains a set of extensible GUI Components to ease the developer's life to create JAVA based Front End/GUI Applications. It is built on top of AWT API and acts as a replacement of AWT API, since it has almost every control corresponding to AWT controls.

Swing API :

* Provides a pluggable look and feel
* Offer a wide variety of controls (buttons, text fields, sliders, color pickers, trees, etc)
* Is light weight. It does not use native code resources unlike AWT

Following is the hierarchy of the classes present in the Java Swing API



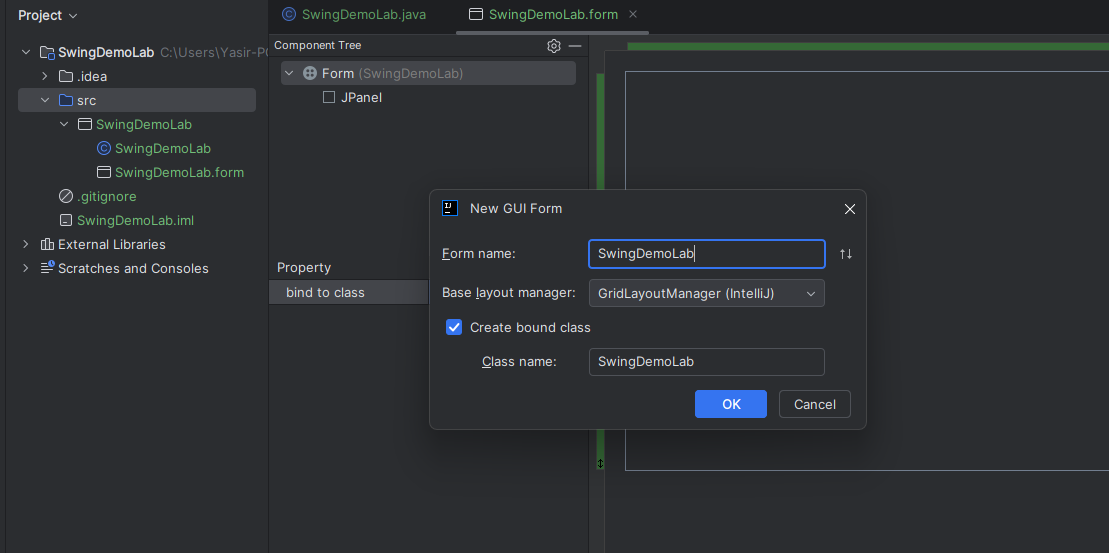
**TUTORIAL**

In this manual we first learn how to create a simple desktop application following a step-by-step approach.

Creating Your First Swing GUI APP in Inteli j

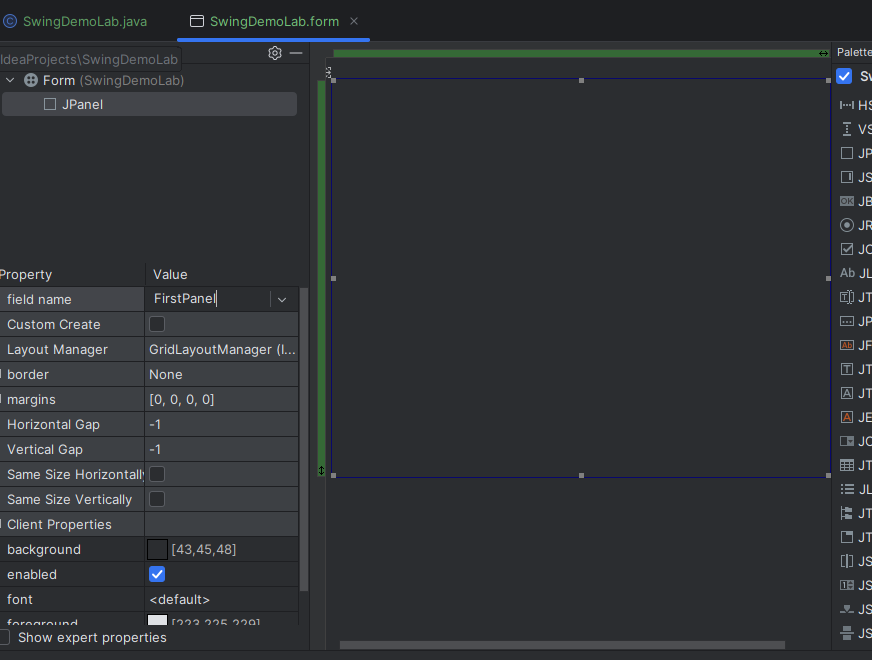
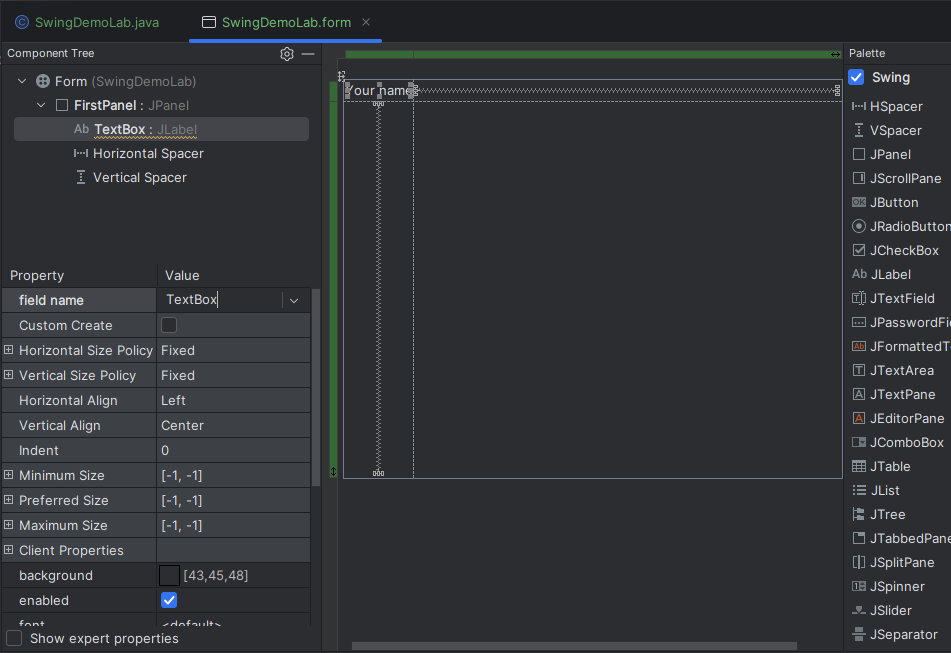
Follow the instructions bellow

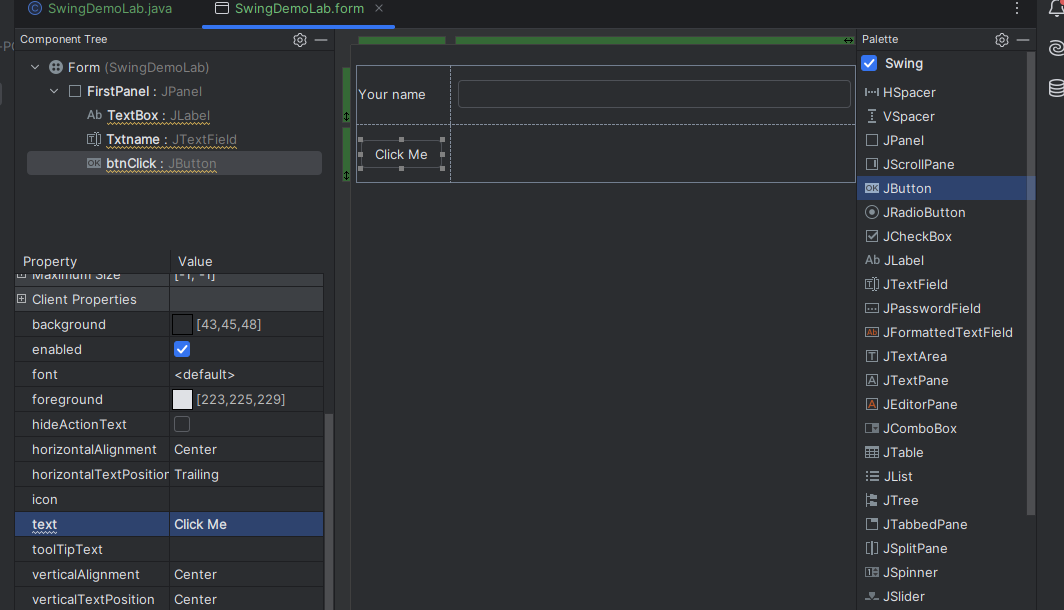
1. File -> New -> Project -> New Project -> Name Your Project and Save it
2. Click on src -> New - > swing UI Designer -> GUI form
3. Provide Your Form Name and chose layout you can chose any but for now just go with default grid layout



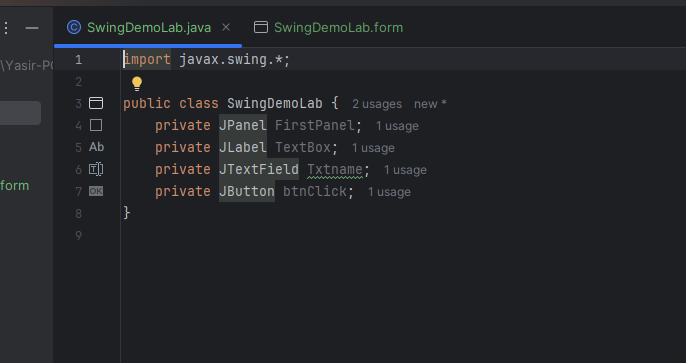
1. After this you will see two different files one with .java Extension and other one with .form
2. .Java will contain all your Code and logic part while the .form will contain all design components.

**Simple Application in swing**

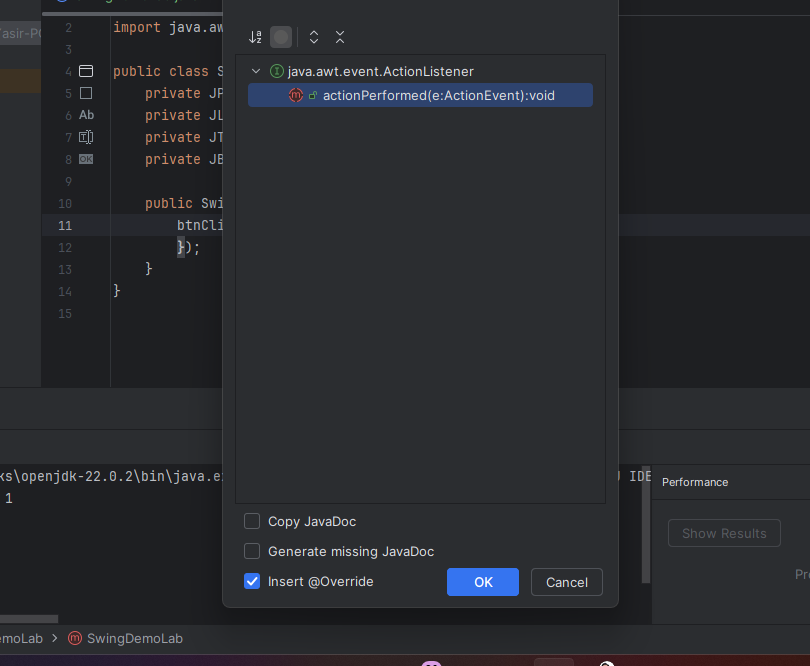
1. Click on the JPanel and give name to your Panel
2. Now lets add some controls/components on that panel I am adding Jlabel here
3. You can change the name and Properties of components/controls from properties menu ad you can change text by changing the text property.
4. Now lets add Another Component that is Textfield place the control and give appropriate name to the Textfield as it will be used in coding part later
5. Now lets add Jbutton and give it appropriate name as it will be use in coding part later and lets text change text property to click me



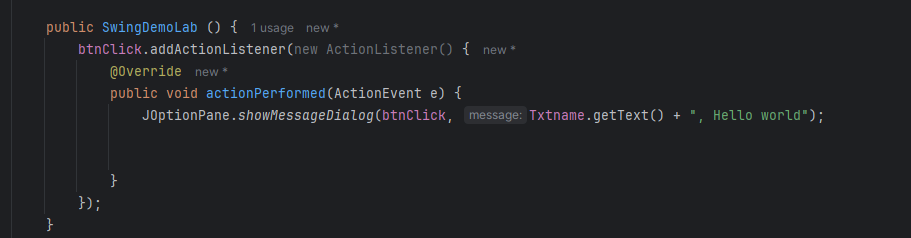
1. Now if we check .java file all components object will be created by default there.



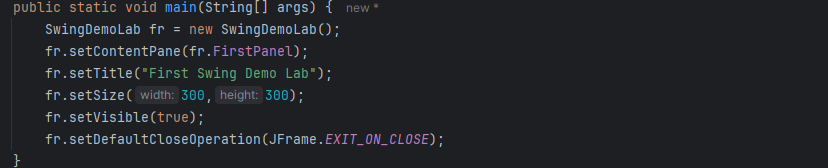
1. Now we will code for the event of button click
2. Right click on the button and go to Option create listner and the select Action listenr and the click ok.



1. After adding the action performed lister add the following code in the method



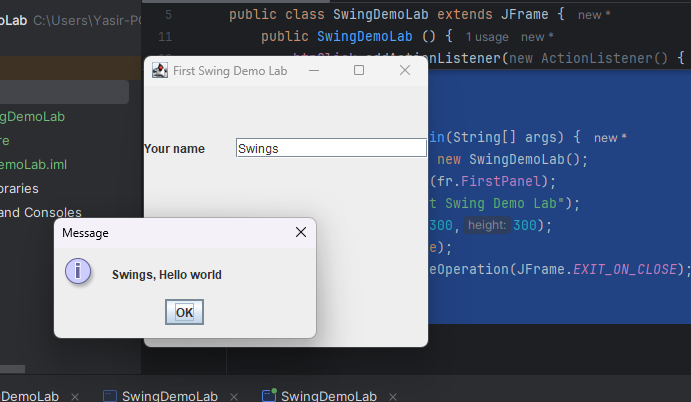
1. Also Extends the Class by Jframe
2. Now to run the code you will neeed main method so include the following code tou your .java file



**Complete code**

import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class SwingDemoLab extends JFrame {  
 private JPanel FirstPanel;  
 private JLabel TextBox;  
 private JTextField Txtname;  
 private JButton btnClick;  
  
 public SwingDemoLab () {  
 btnClick.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JOptionPane.*showMessageDialog*(btnClick, Txtname.getText() + ", Hello world");  
  
  
 }  
 });  
 }  
 public static void main(String[] args) {  
 SwingDemoLab fr = new SwingDemoLab();  
 fr.setContentPane(fr.FirstPanel);  
 fr.setTitle("First Swing Demo Lab");  
 fr.setSize(300,300);  
 fr.setVisible(true);  
 fr.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 }  
}

1. Now run the application



**Swing Components and Their Descriptions**

| **Component** | **Description** |
| --- | --- |
| JFrame | A top-level container used to create a window. |
| JPanel | A generic container used to hold and organize components. |
| JLabel | A display area for a short text string or an image, or both. |
| JTextField | A single-line text field for user input. |
| JButton | A button that triggers an action event when clicked. |
| JCheckBox | A component that can be selected or deselected, representing a boolean choice. |
| JRadioButton | A button that can be selected or deselected, typically used in groups to allow one selection. |
| JComboBox | A drop-down list that allows users to choose one item from a list. |
| JTextArea | A multi-line area to display or input text. |
| JPasswordField | A text field that hides the input, typically used for password entry. |
| JTable | A component that displays data in a two-dimensional table format. |
| JScrollPane | A container that provides a scrollable view of another component. |
| JMenuBar | A menu bar that can be added to the top of a JFrame to provide options for users. |