

ITSE322 Modern Programming  
Language:  
Advanced Java

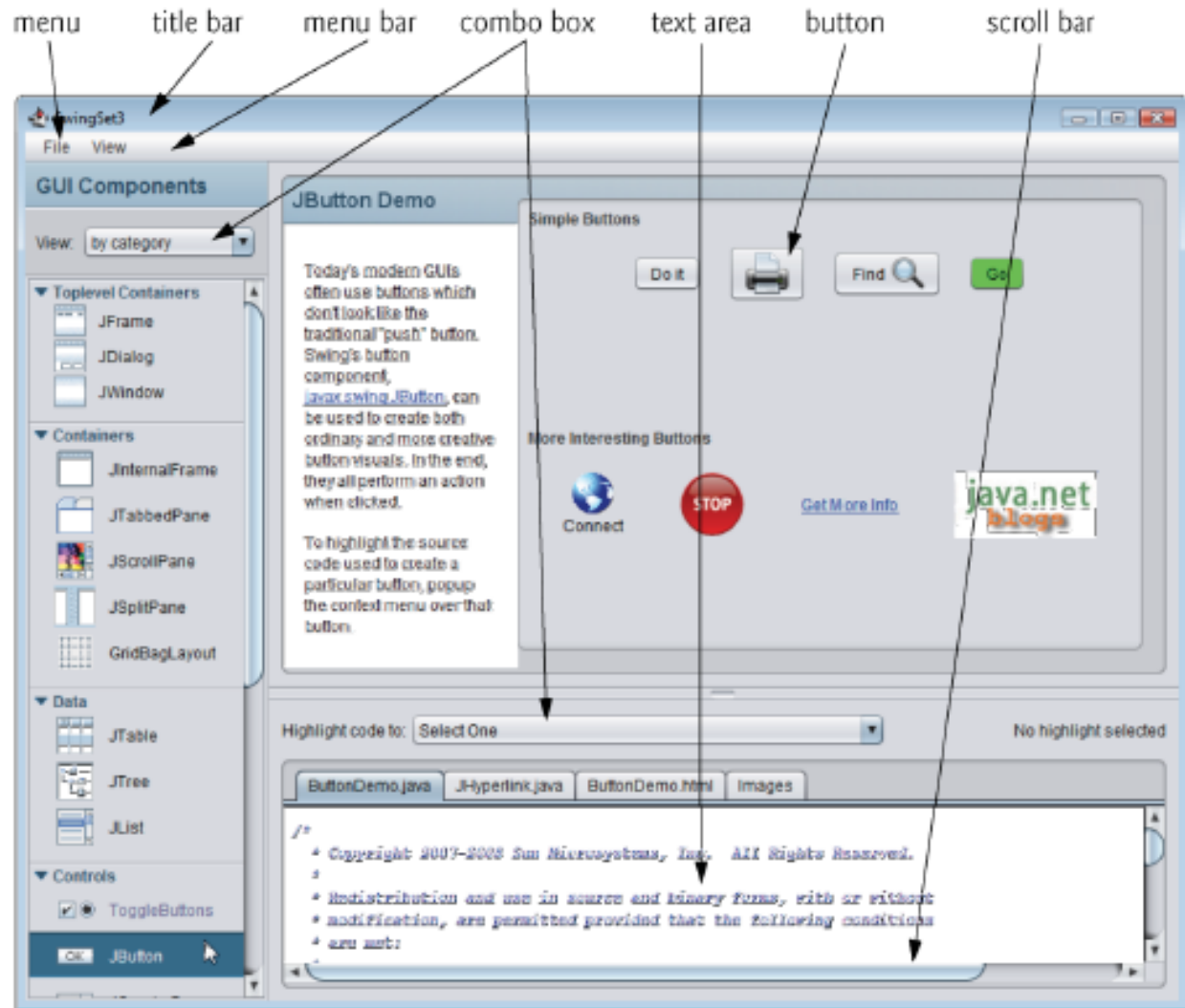
Java GUI  
Lecture 5

# Learning Objectives

1. Create simple graphical user interfaces (GUI's) in Java
2. Learn about event-driven model
3. Build GUI for your database

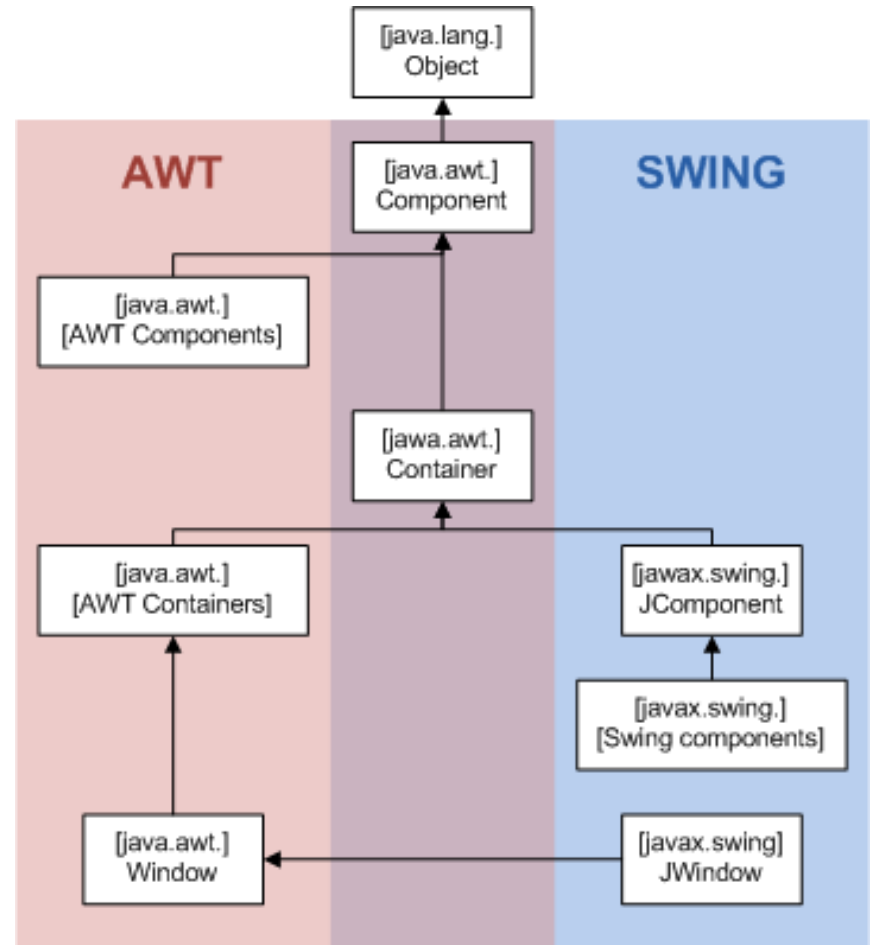
# What's in a GUI?

Answer:  
A bunch of  
graphical  
objects!



# The Java GUI framework

- ▶ Abstract Windowing Toolkit (AWT)
  - Built on the native OS
  - Faster
  - Can be used in browsers without a java plugin
- Swing
  - Newer – built on AWT.
  - Made completely in Java
  - More Portable
  - Easier to use
  - Can use the ‘Model View Control’ design process

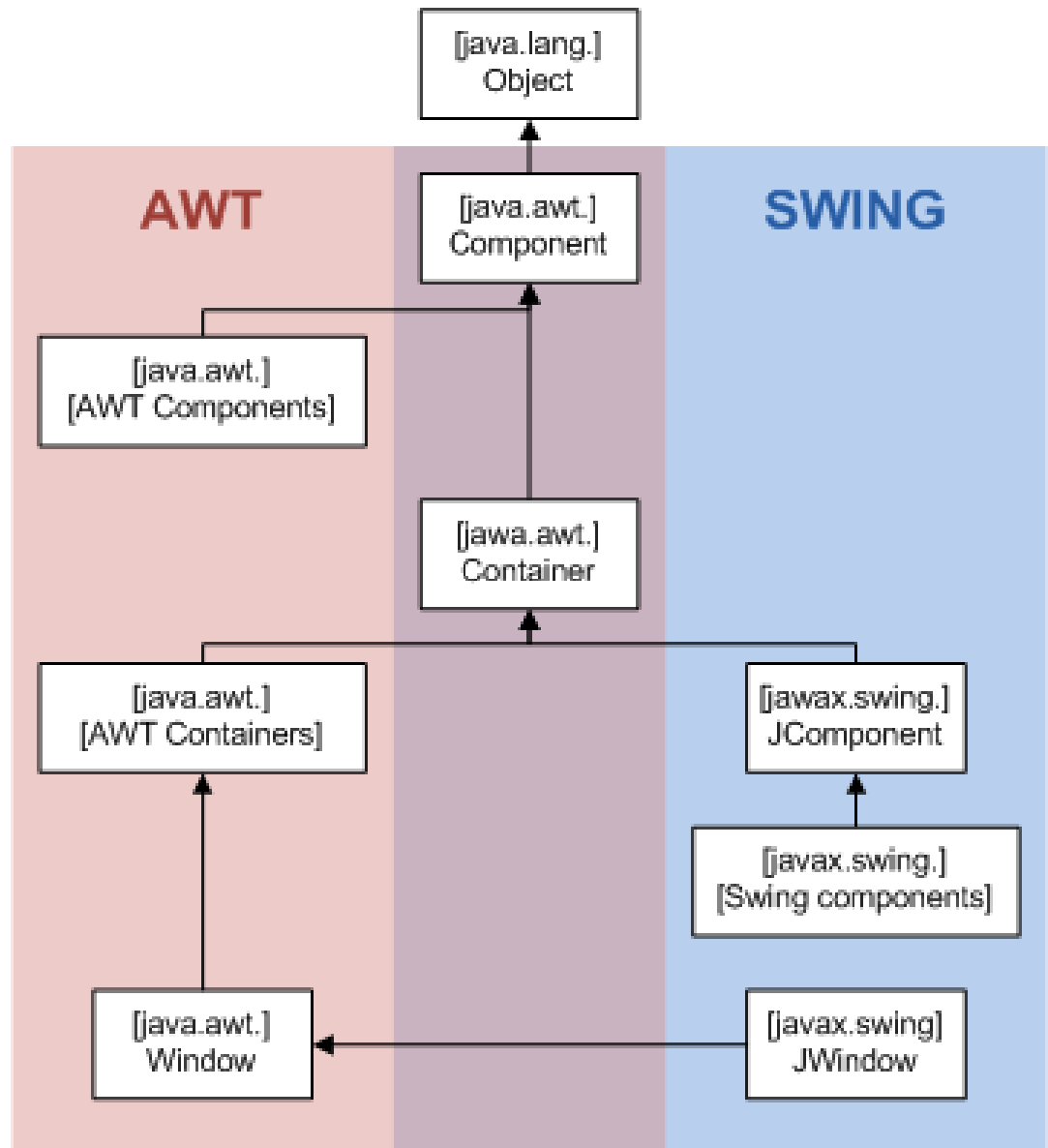


API:

<http://java.sun.com/j2se/1.3/docs/api/index.html>

# Swing

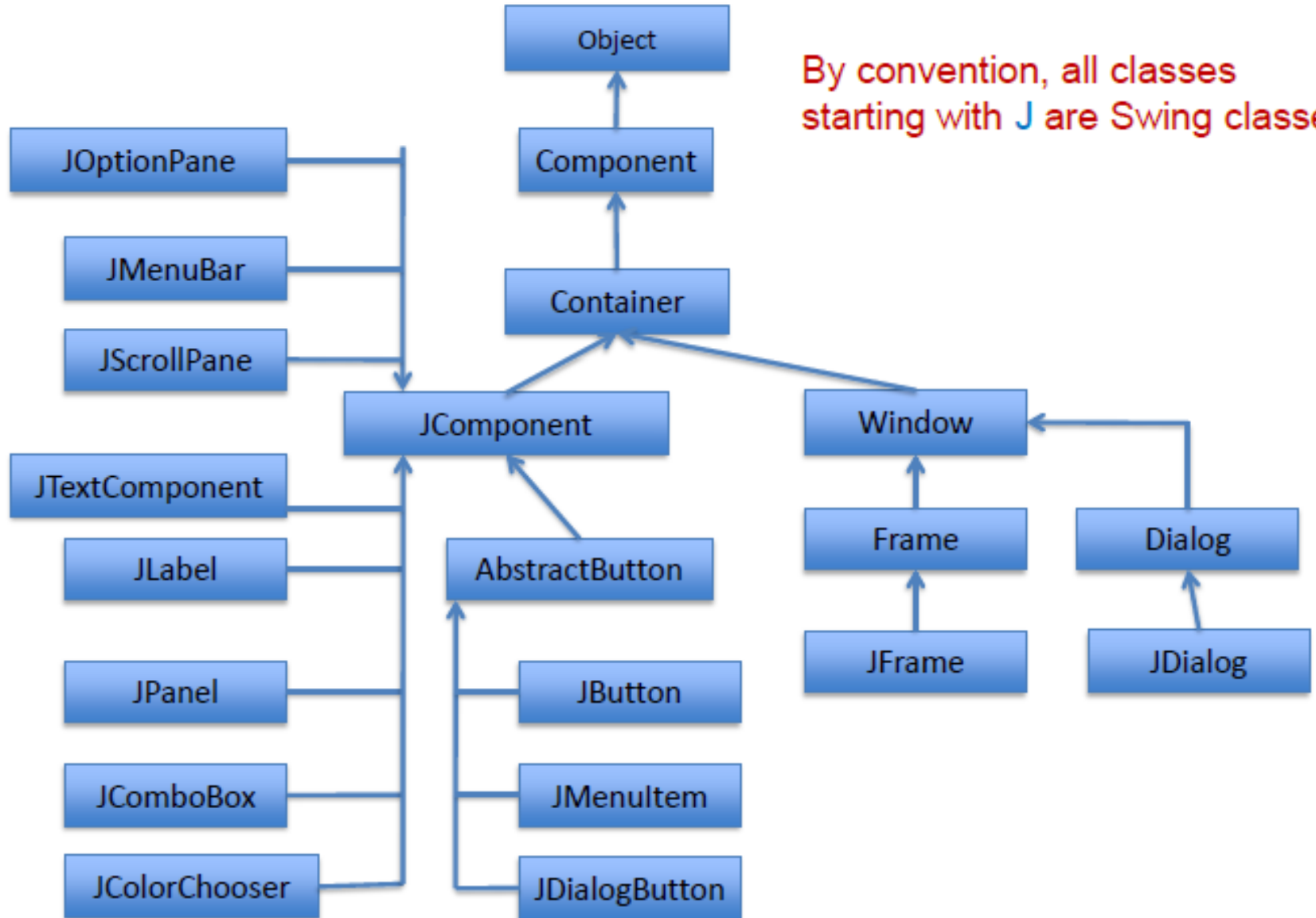
- The JComponent class is the root of the swing component hierarchy
  - All swing components are subtypes of this except for top-level containers such as JFrame



# More Swing Components

Component	Description
JLabel	An area that can display text
JTextField	An area in which the user may type a single line of input from the key board
JComboBox	A component that displays a drop-down list of items from which the user may select. A combo box also provides a text field in which a use may type input. It is a combo box as it is a combination of a list and a text field
JCheckBox	A component that has a box that may be checked or unchecked
List	A list from which a user may select an item
JRadioButton	A control that can be either selected or deselected. Radio buttons usually appear in groups and allow the user to select one of several options
JSlider	A control that allows the user to select a value by moving a slider along a track
JButton	A button that can cause an action to occur when clicked

# Hierarchical View



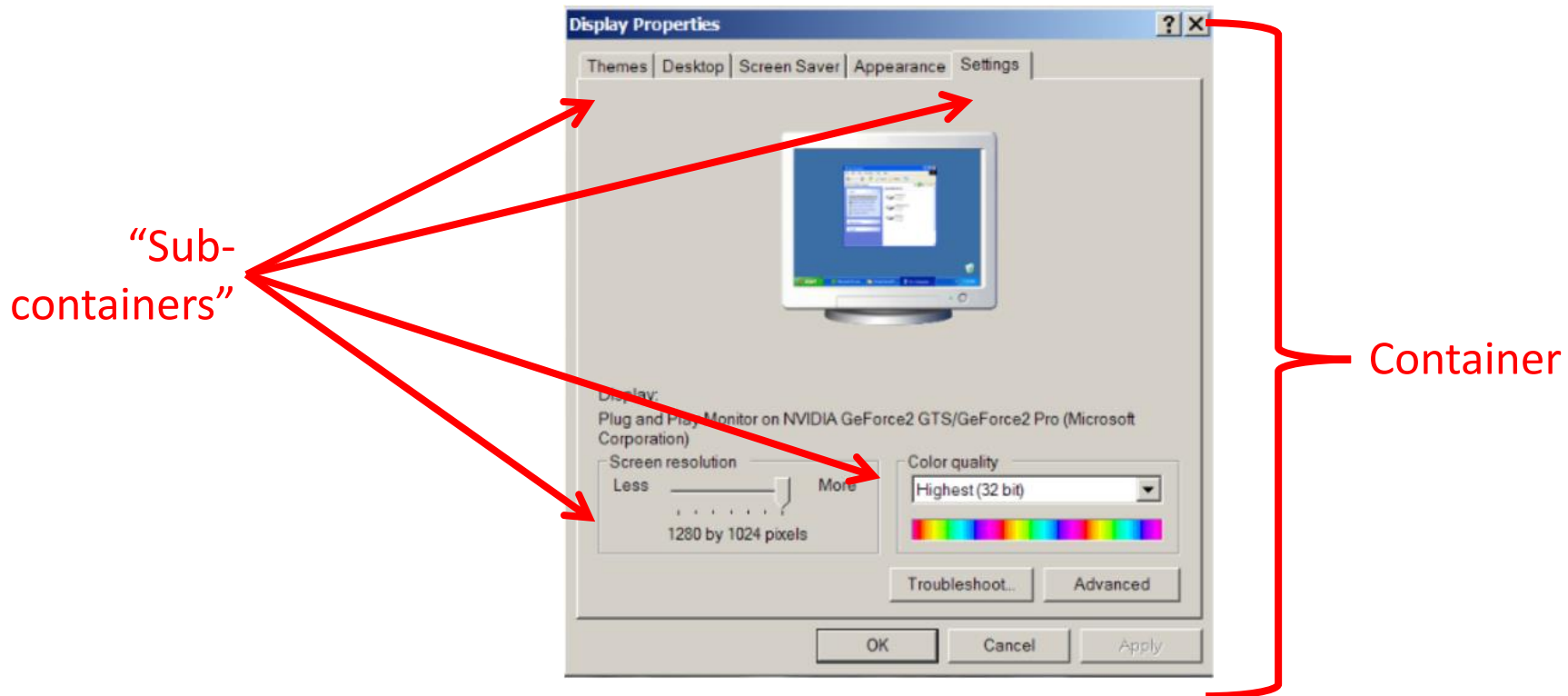
# Components

- There are many types of graphical controls and displays available:
  - JButton, JFrame, JLabel, JList, JTextArea, Window
- A graphical component is also known as a “widget”



# Containers

- A special type of Component that is used to hold other components.
- Can be used to group components on the screen (i.e., one container holds another container which in turn groups a number of controls).



# GUI Component API

- Java: GUI component = class

- Properties

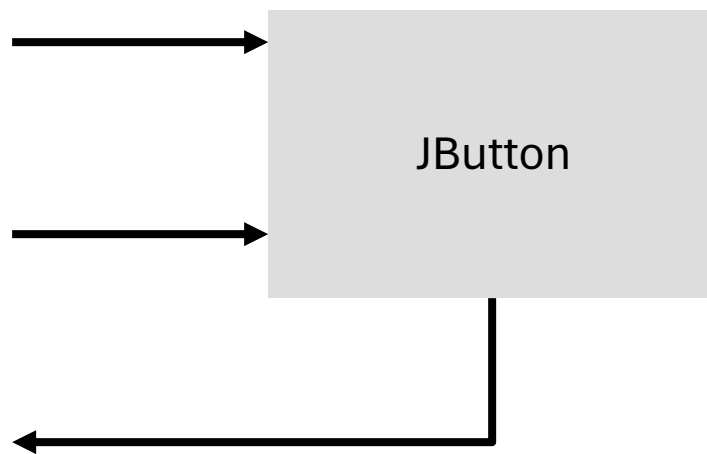
- 

- Methods

- 

- Events

- 



# Using a GUI Component

## 1. Create it

- Instantiate object: `b = new JButton("press me");`

## 2. Configure it

- Properties: `b.text = "press me";` [avoided in java]
- Methods: `b.setText("press me");`

## 3. Add it

- `panel.add(b);`

## 4. Listen to it

- Events: Listeners

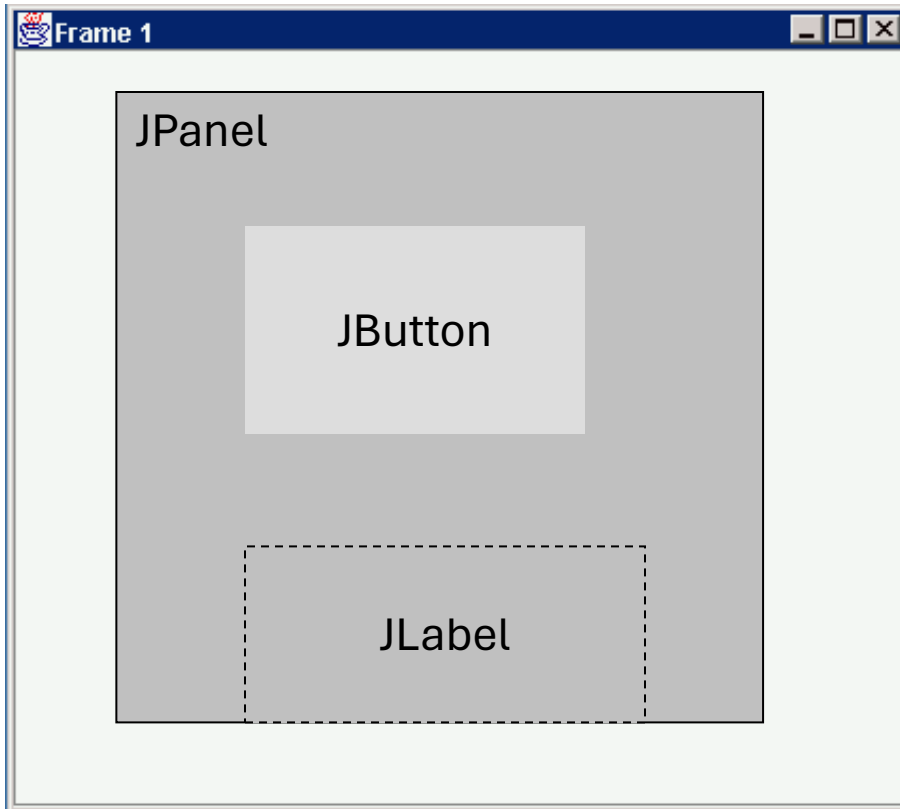


JButton

# Anatomy of an Application GUI

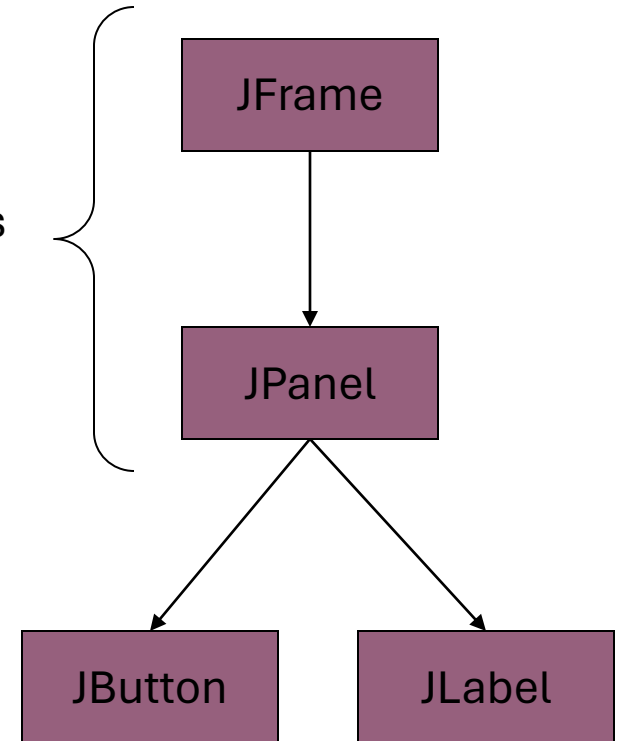
GUI

JFrame



containers

Internal structure



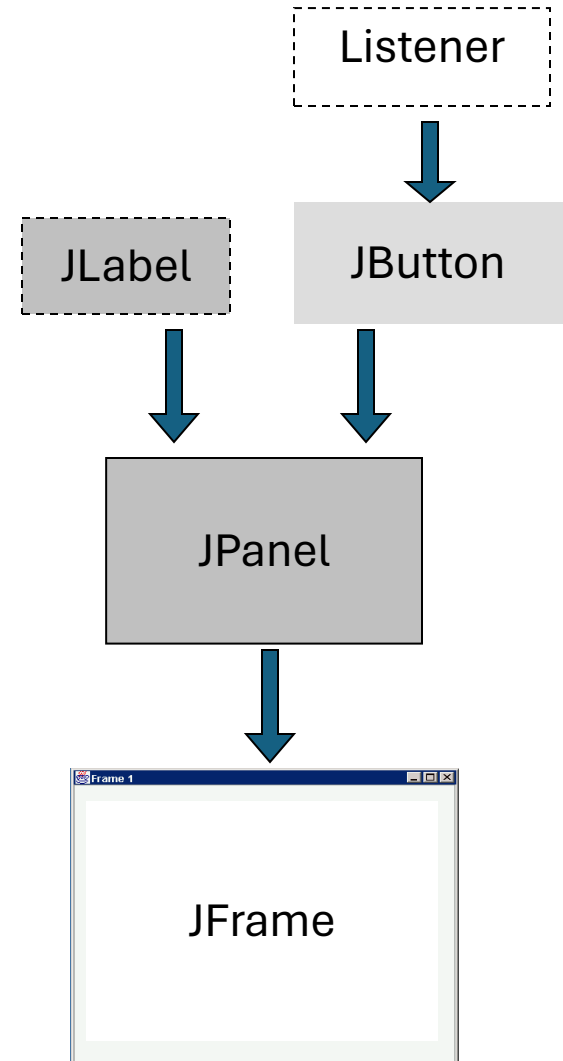
# Build from bottom up

- Create:

- Frame
- Panel
- Components
- Listeners

- Add:

- listeners into components
- components into panel
- panel into frame



# Code

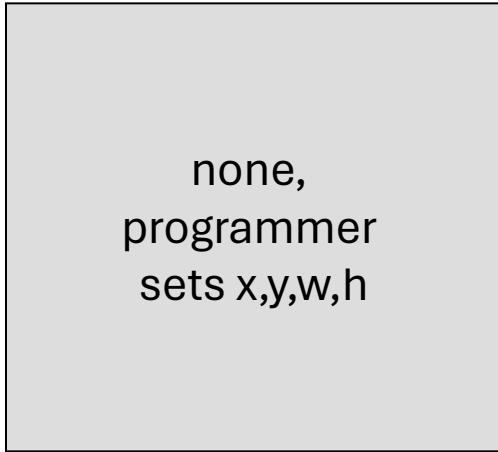
```
import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class SimpleGUI1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("TITLE");
        //1. Create it
        JPanel panel = new JPanel();
        JButton button = new JButton("PRESS ME");
        //2. Configure it
        frame.setTitle("My Frame");
        frame.setSize(400,100);
        button.setBackground(Color.YELLOW);
        //3. add it
        panel.add(button); // add button to panel
        frame.setContentPane(panel); // add panel to frame
        frame.setVisible(true);
    }
}
```

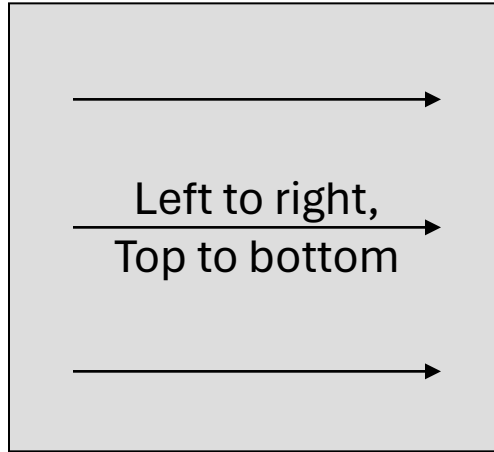


# Layout Manager Heuristics

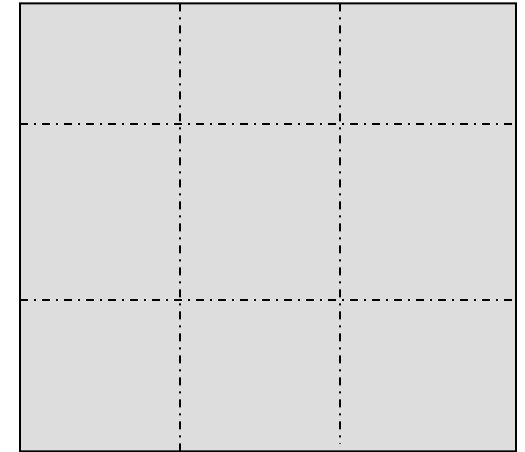
null



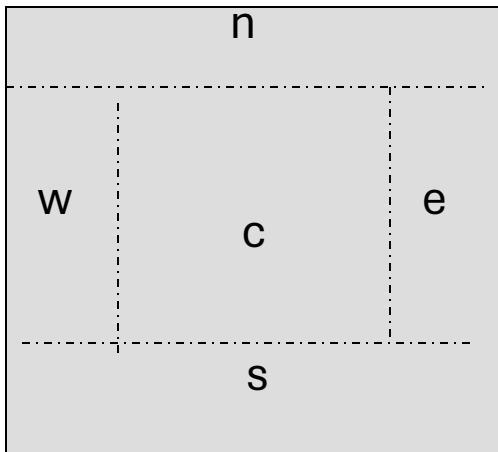
FlowLayout



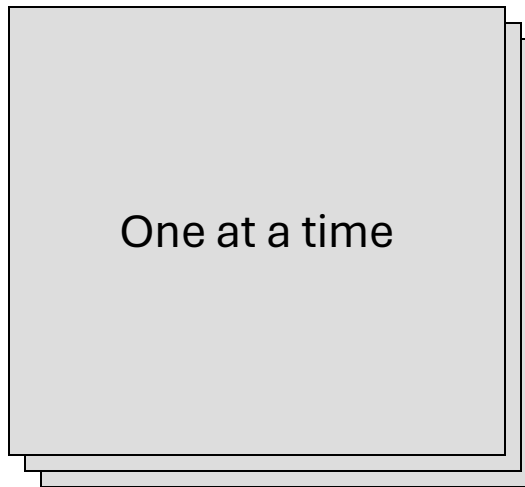
GridLayout



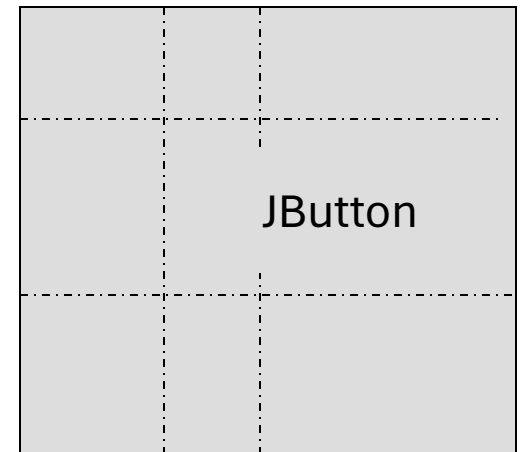
BorderLayout



CardLayout

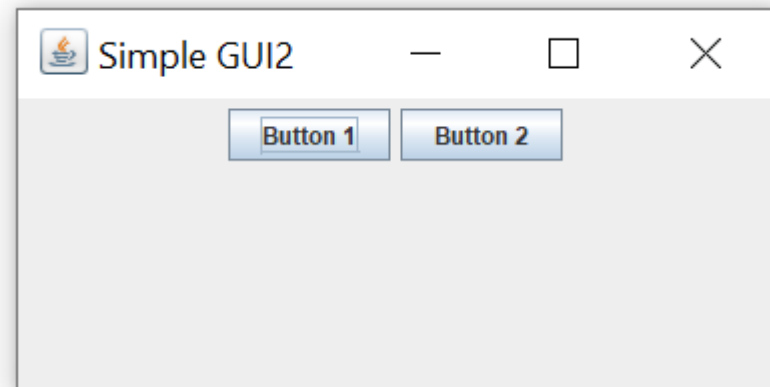


GridBagLayout



# Flow Layout

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Container;
import java.awt.FlowLayout;
class SimpleGUI2 extends JFrame{
    public SimpleGUI2()
    {
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        //add button
        JButton but1 = new JButton("Button 1");
        JButton but2 = new JButton("Button 2");
        Container cp = getContentPane();//must do this
        cp.setLayout(new FlowLayout());
        cp.add(but1);
        cp.add(but2);
        setTitle("Simple GUI2");
        setVisible(true);
    }
    public static void main(String[] args)
    {
        SimpleGUI2 gui = new SimpleGUI2();
        gui.setSize(400,200);
    }
}
```





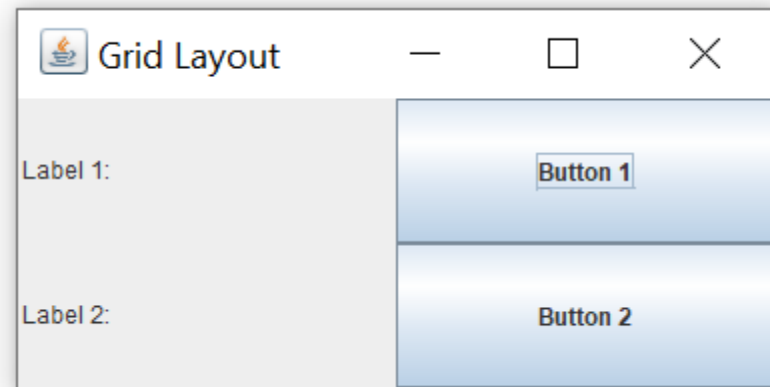
# Grid Layout

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.Label;

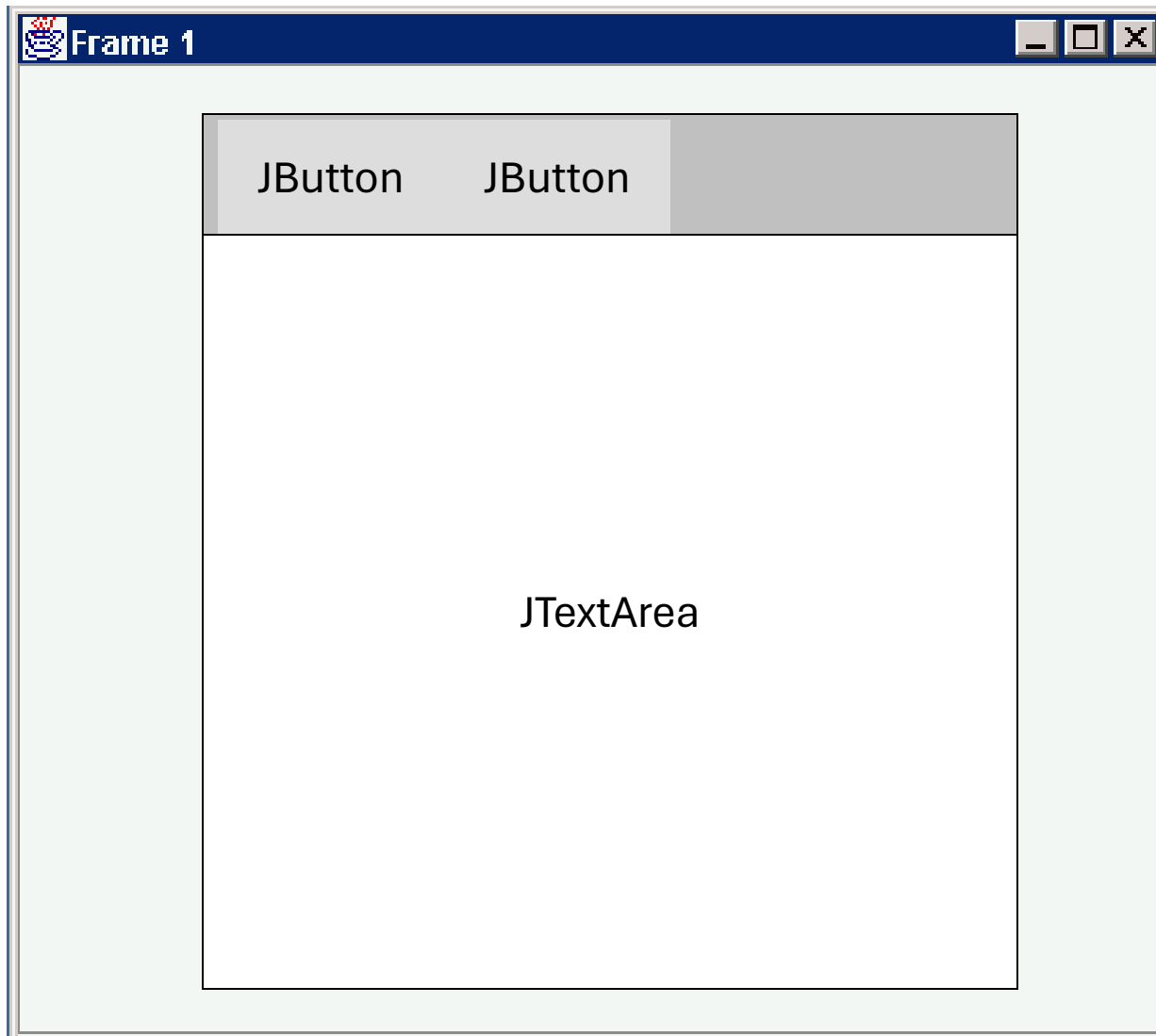
class SimpleGUI3 extends JFrame{
    public SimpleGUI3()
    {
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        //add button
        JButton but1 = new JButton("Button 1");
        JButton but2 = new JButton("Button 2");
        Container cp = getContentPane();//must do this
        cp.setLayout(new GridLayout(2,2));
        cp.add(new Label("Label 1:"));
        cp.add(but1);
        cp.add(new Label("Label 2:"));
        cp.add(but2);

        setTitle("Grid Layout ");
        setVisible(true);
    }

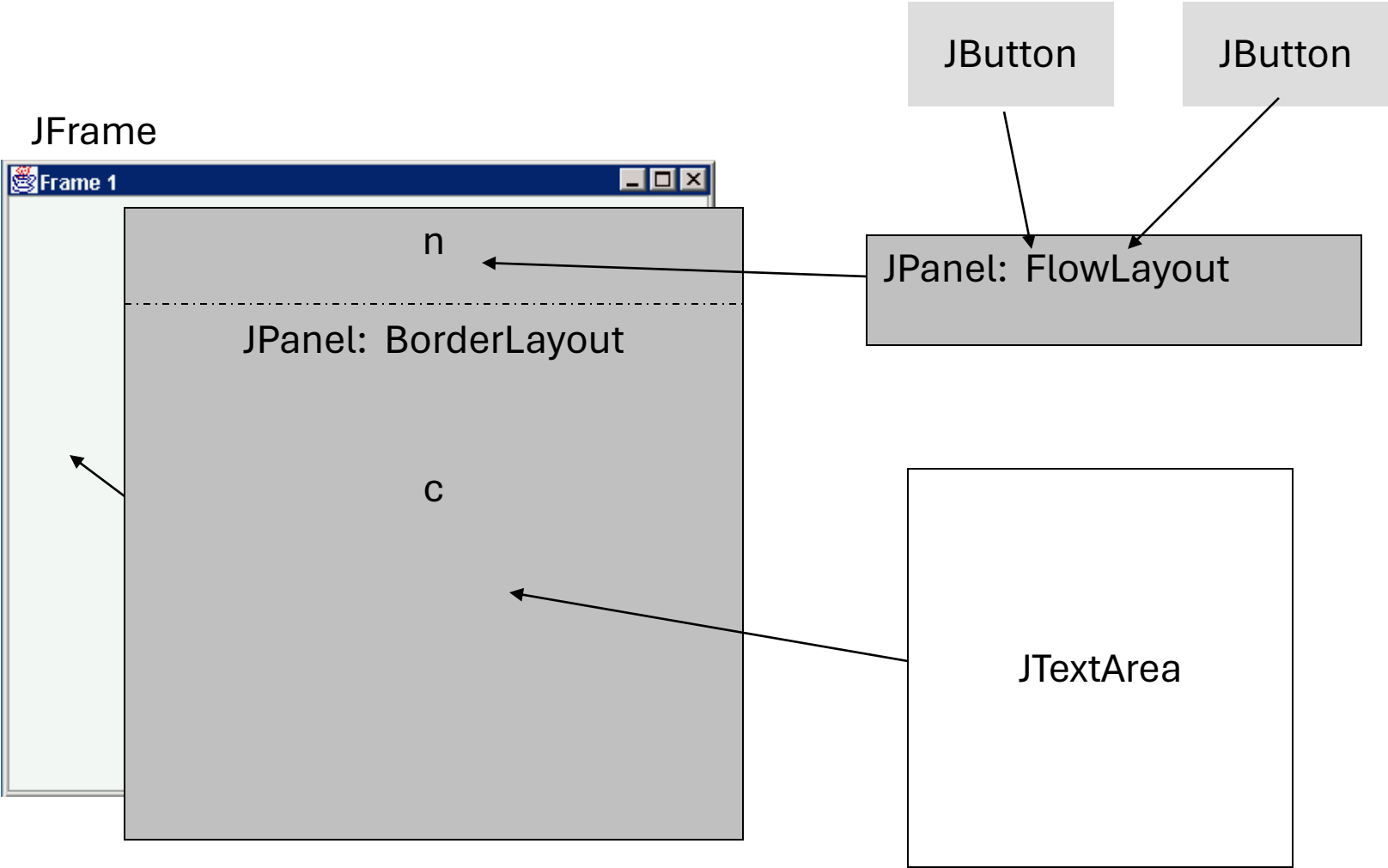
    public static void main(String[] args)
    {
        SimpleGUI3 gui = new SimpleGUI3();
        gui.setSize(400,200);
    } }
```



# Combinations



# Combinations



# Action Listeners

```
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.JOptionPane;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

class SimpleGUI4 extends JFrame{
    private JButton but1;
    private JButton but2;
    public SimpleGUI4()
    {
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        //add button
        but1 = new JButton("Click me");
        but2 = new JButton("Exit");
        Container cp = getContentPane();//must do this
        cp.setLayout(new FlowLayout());
        MyActionListener al = new MyActionListener();
        but1.addActionListener(al);
        but2.addActionListener(al);
        cp.add(but1);
        cp.add(but2);
        setTitle("Simple GUI4");
        setVisible(true);
    }
}
```

```
public class MyActionListener implements ActionListener{
    public void actionPerformed(ActionEvent e){
        Object source = e.getSource();
        if(source == but1){
            JOptionPane.showMessageDialog(null, "Close ME!");
        } else if( source == but2)
        {
            System.exit(0);
        }
    }
}

public static void main(String[] args)
{
    SimpleGUI4 gui = new
SimpleGUI4();
    gui.setSize(400,200); //set frames size in pixels
}
}
```

