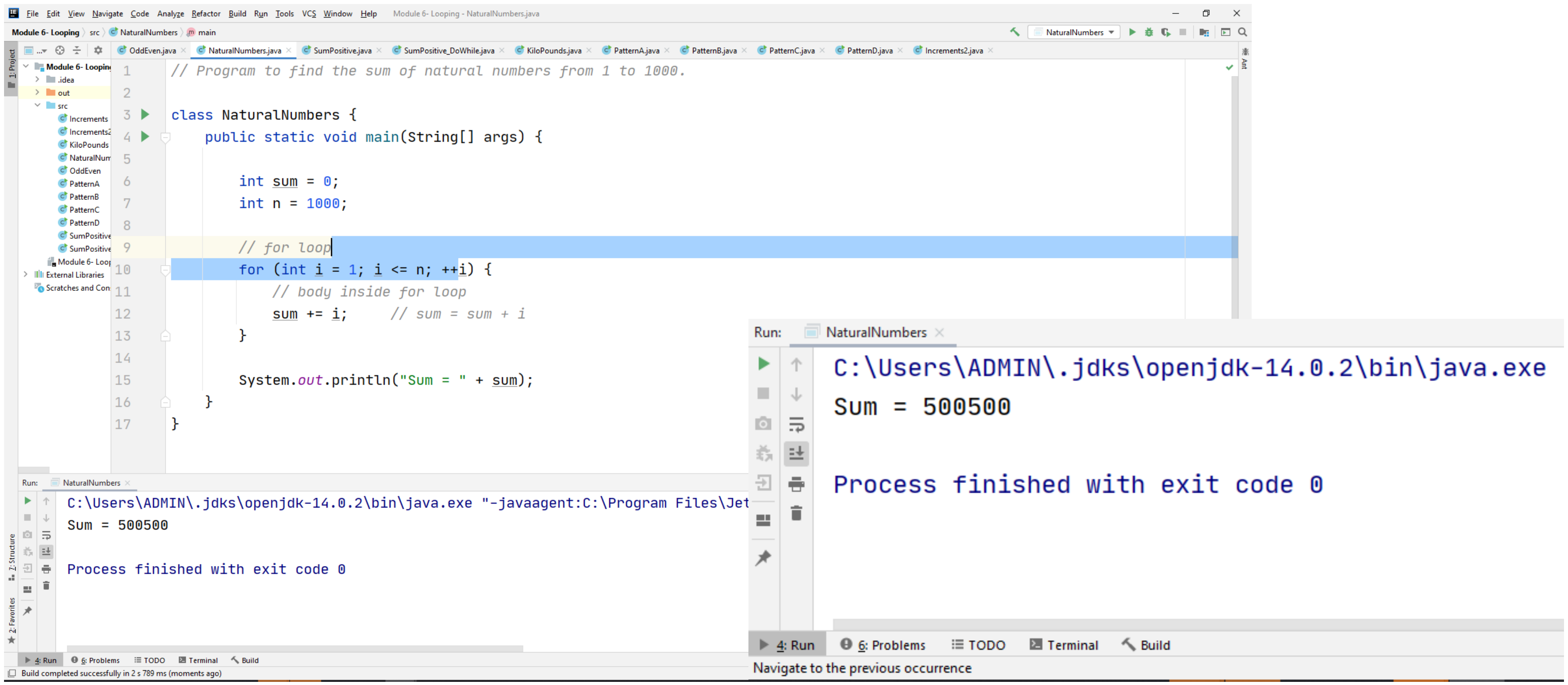
NCP2103: Object-Oriented Programming (Java Programming)

Graphical User Interface - Swing

Errol John M. Antonio
Assistant Professor
Computer Engineering Department
University of the East – Manila Campus

Module Outline

I: Introduction



```java
// Program to find the sum of natural numbers from 1 to 1000.

class NaturalNumbers {
    public static void main(String[] args) {

        int sum = 0;
        int n = 1000;

        // for loop
        for (int i = 1; i <= n; ++i) {
            // body inside for loop
            sum += i;     // sum = sum + i
        }

        System.out.println("Sum = " + sum);
    }
}
```

```
C:\Users\ADMIN\.jdks\openjdk-14.0.2\bin\java.exe "-javaagent:C:\Program Files\Jet
Sum = 500500

Process finished with exit code 0
```

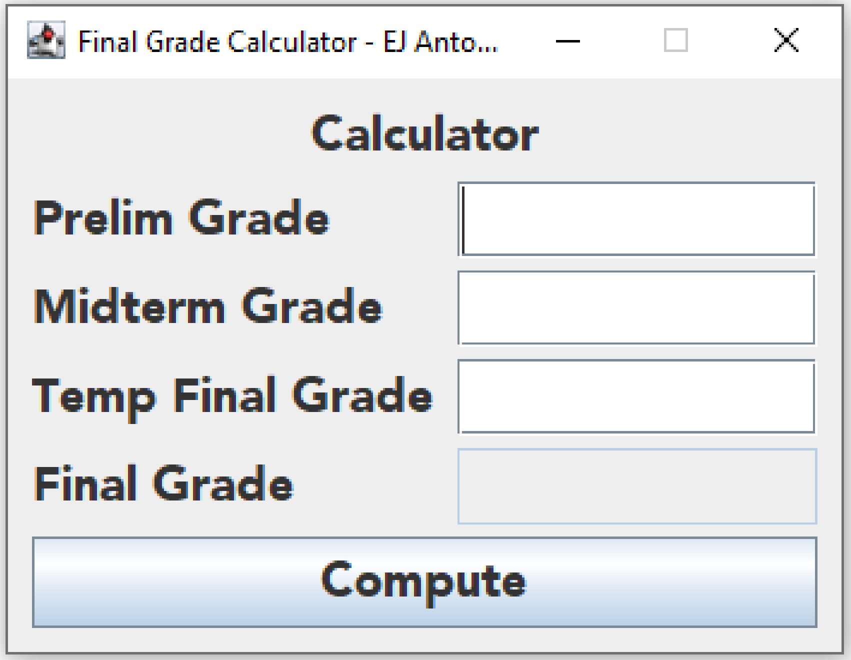
```
C:\Users\ADMIN\.jdks\openjdk-14.0.2\bin\java.exe
Sum = 500500

Process finished with exit code 0
```

Build completed successfully in 2 s 789 ms (moments ago)

Navigate to the previous occurrence

Command Line Interface (CLI) / Console

Introduction



Graphical User Interface (GUI)

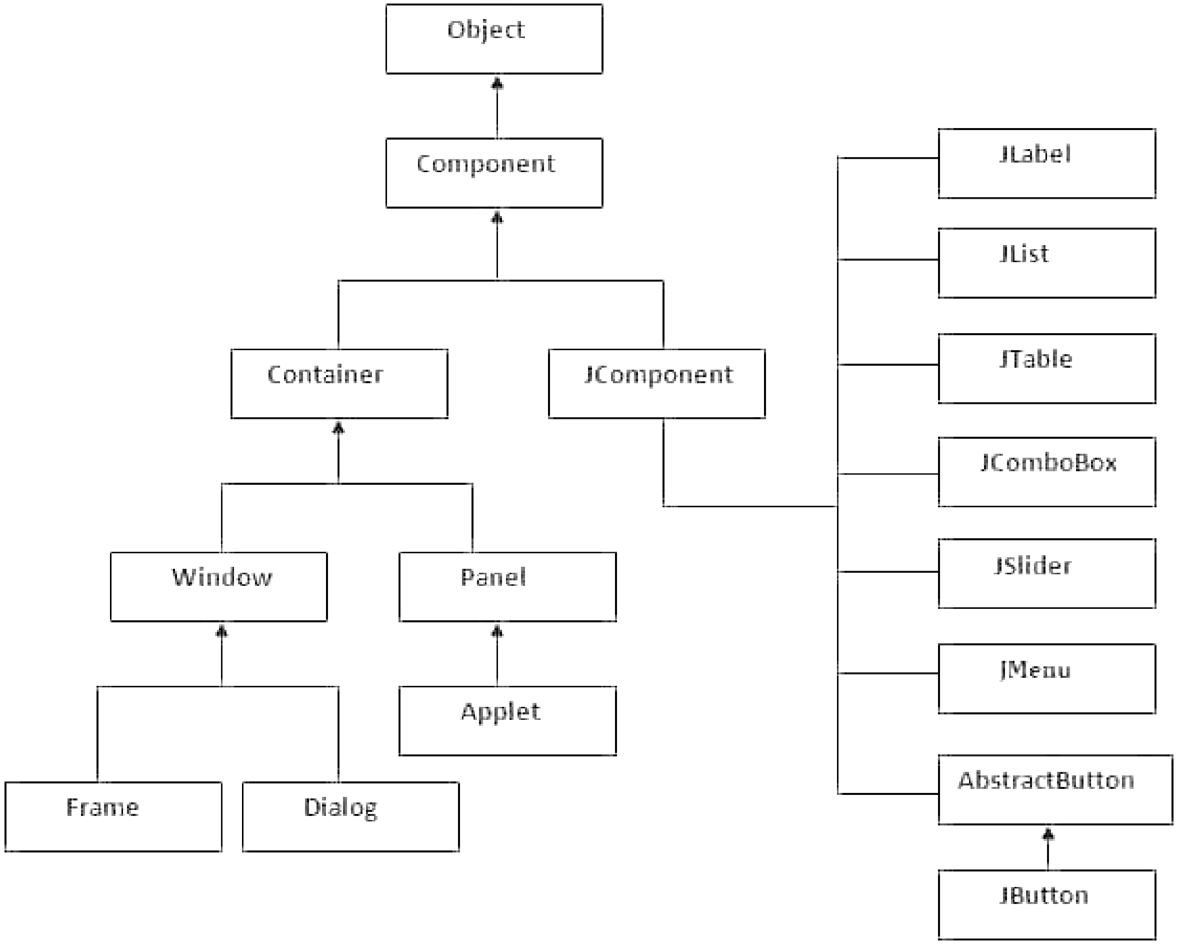
Introduction

- Java APIs for GUI Programming
  - AWT (Abstract Window Toolkit)
  - Swing
  - JavaFX

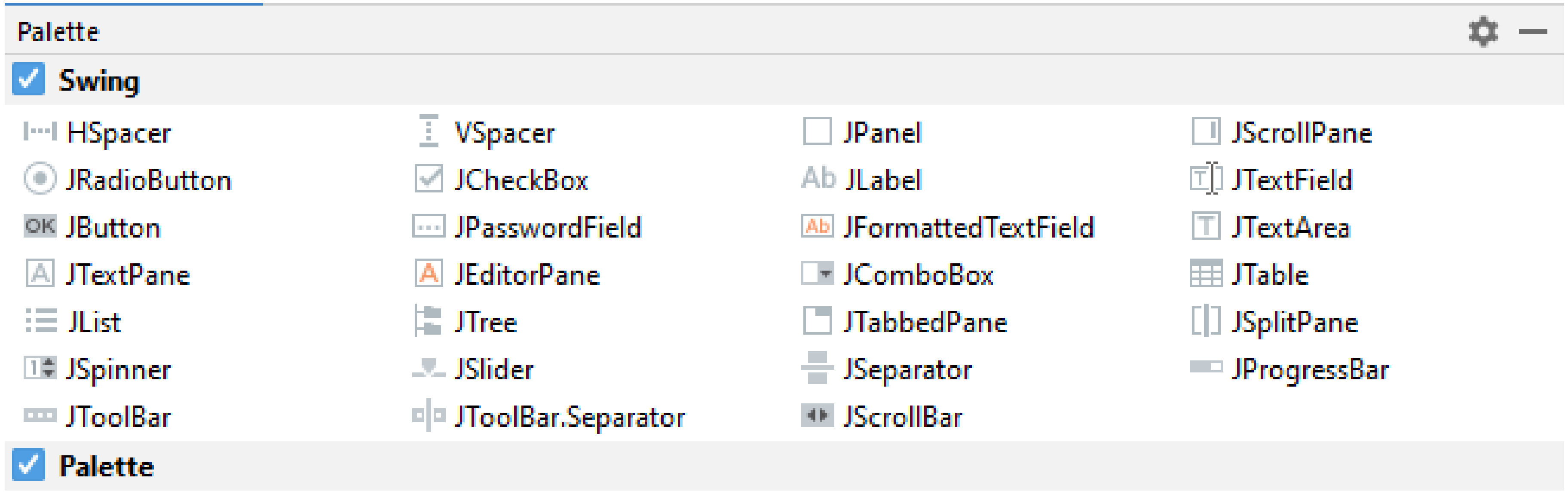
Introduction: AWT vs Swing

AWT	Swing
AWT components are **platform-dependent.**	Java swing components are **platform-independent.**
AWT components are **heavyweight.**	Swing components are **lightweight.**
AWT **doesn't support pluggable look and feel.**	Swing **supports pluggable look and feel.**
AWT provides **less components** than Swing.	Swing provides **more powerful components** such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
AWT **doesn't follows MVC**(Model View Controller)	Swing **follows MVC.**

Swing Components Overview



Swing Components Overview



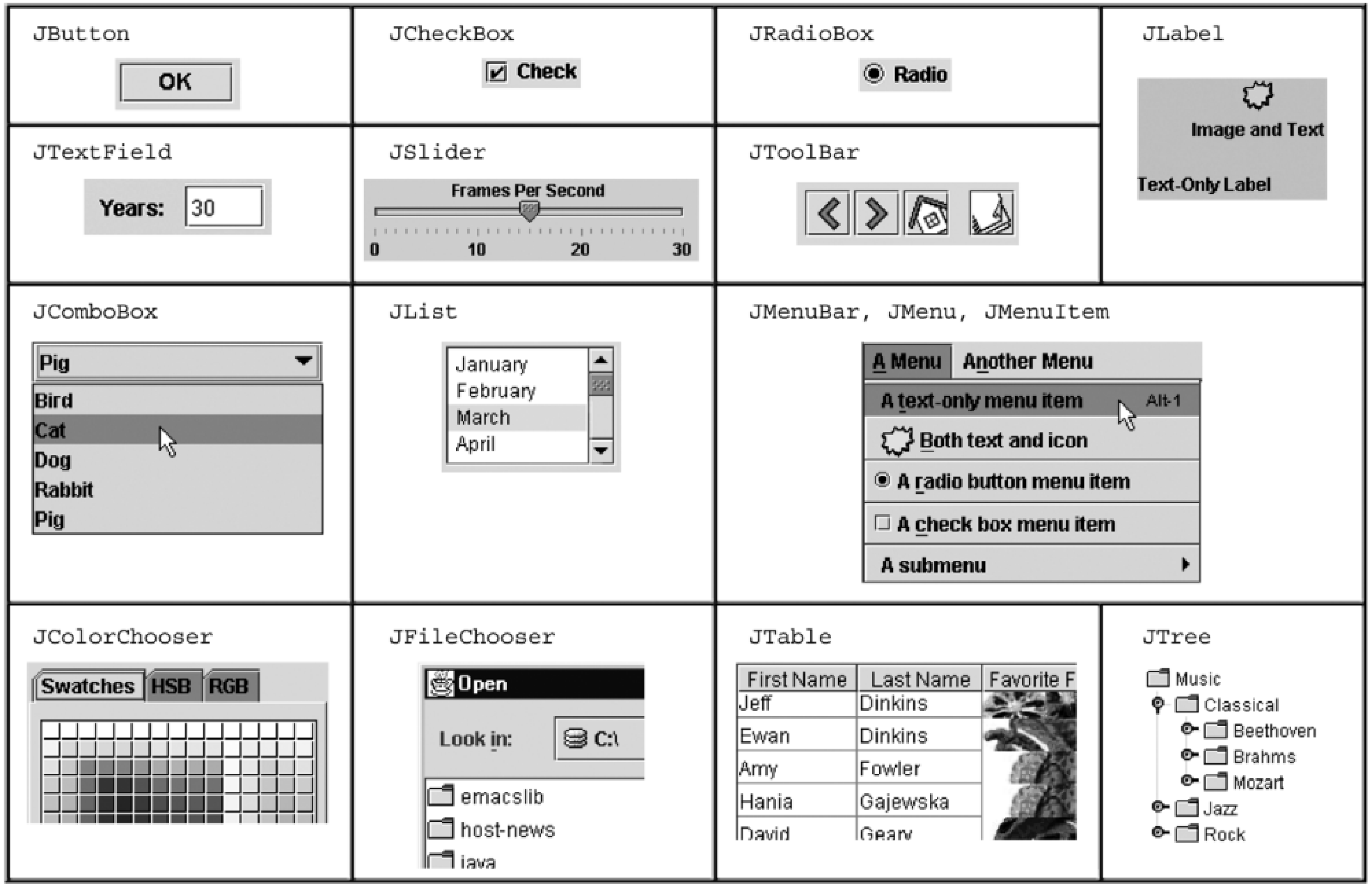
Palette

✓ Swing

HSpacer	VSpacer	JPanel	JScrollPane
JRadioButton	JCheckBox	JLabel	JTextField
JButton	JPasswordField	JFormattedTextField	JTextArea
JTextPane	JEditorPane	JComboBox	JTable
JList	JTree	JTabbedPane	JSplitPane
JSpinner	JSlider	JSeparator	JProgressBar
JToolBar	JToolBar.Separator	JScrollBar	

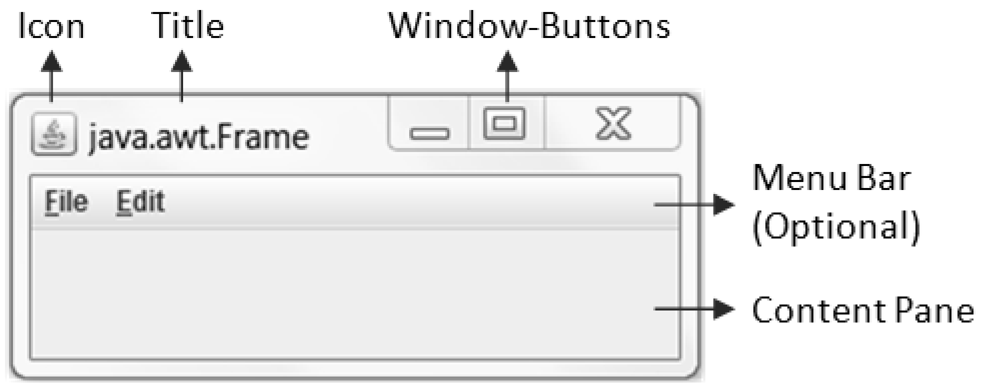
✓ Palette

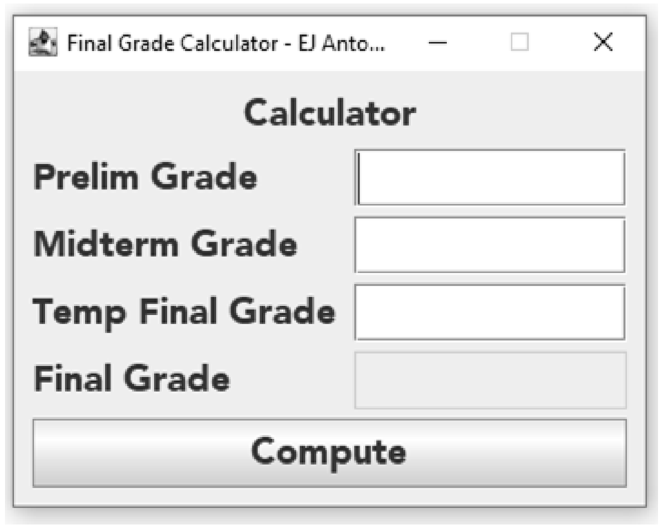
Swing Components Overview

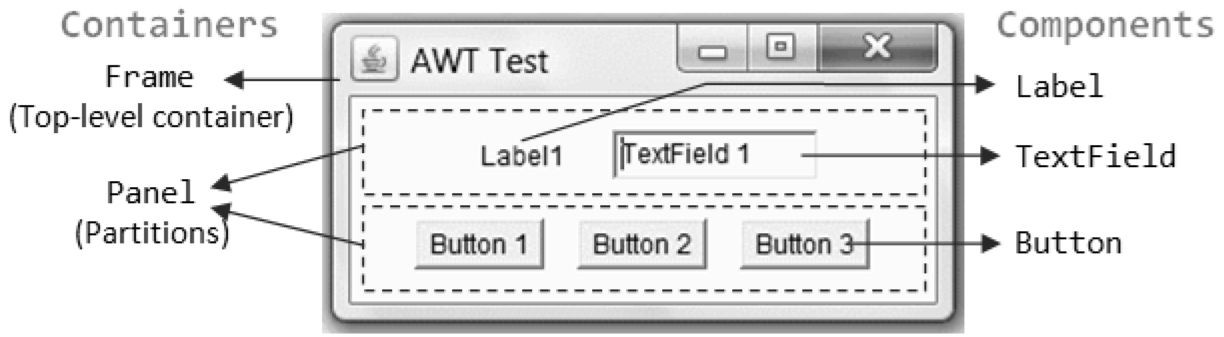


First Name	Last Name	Favorite F
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

II: GUI Basics

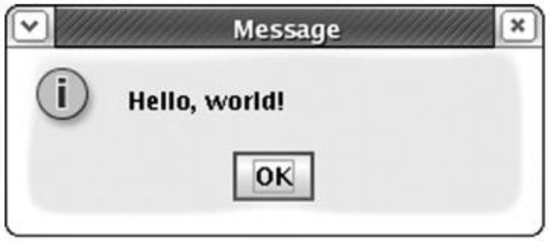




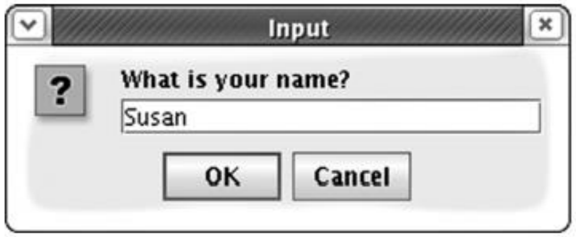


Option Pane

- An option pane is a simple message box that appears on the screen and presents a message or a request for input to the user.
- **`import javax.swing.*;`**

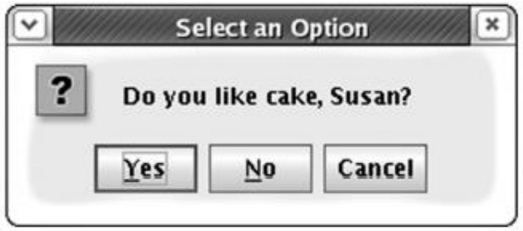

Message

Hello, world!

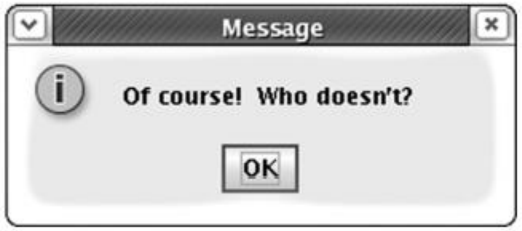

Input

What is your name?

Susan


Select an Option

Do you like cake, Susan?


Message

Of course! Who doesn't?

Option Pane

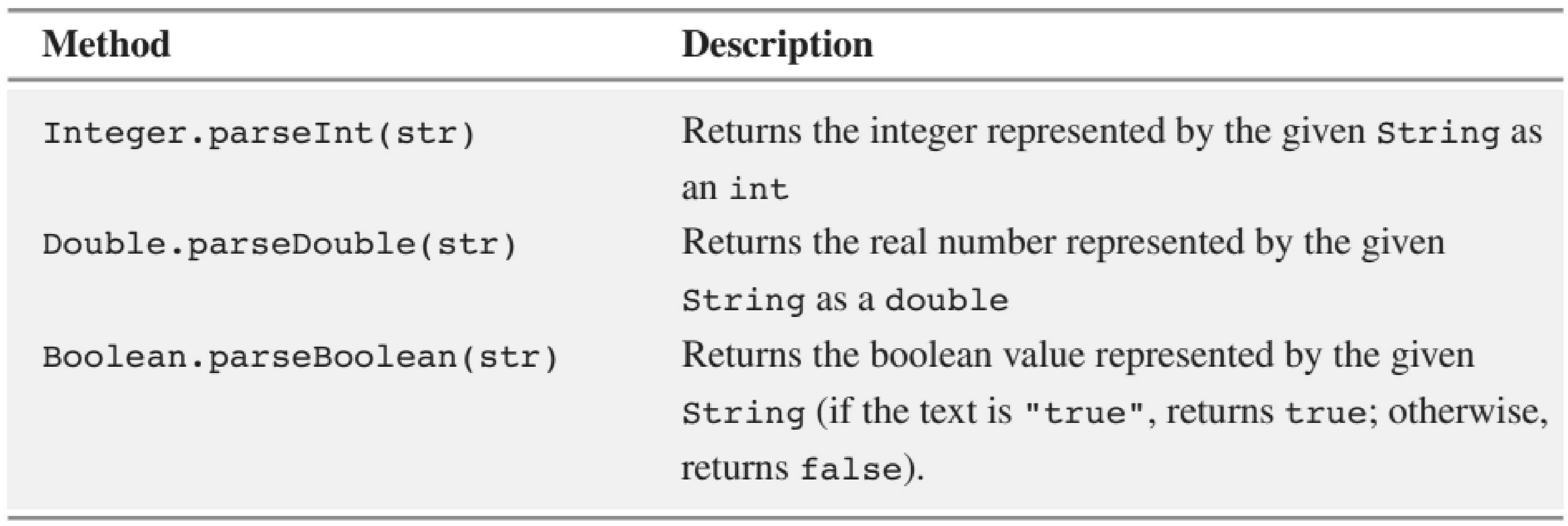
Table 14.1 Useful Methods of the `JOptionPane` Class

Method	Description
`showConfirmDialog(parent, message)`	Shows a Yes/No/Cancel message box containing the given message on the screen and returns the choice as an `int` with one of the following constant values: • `JOptionPane.YES_OPTION` (user clicked "Yes") • `JOptionPane.NO_OPTION` (user clicked "No") • `JOptionPane.CANCEL_OPTION` (user clicked "Cancel")
`showInputDialog(parent, message)`	Shows an input box containing the given message on the screen and returns the user's input value as a `String`
`showMessageDialog(parent, message)`	Shows the given message string in a message box on the screen

Pitfall:

- One limitation of `JOptionPane` is that its `showConfirmDialog` method always returns the user's input as a `String`.

Table 14.2 Useful Methods of Wrapper Classes

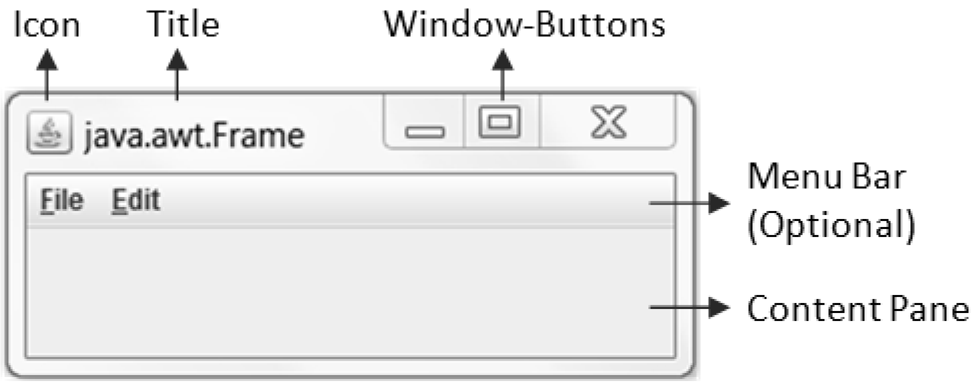


Method	Description
`Integer.parseInt(str)`	Returns the integer represented by the given `String` as an `int`
`Double.parseDouble(str)`	Returns the real number represented by the given `String` as a `double`
`Boolean.parseBoolean(str)`	Returns the boolean value represented by the given `String` (if the text is `"true"`, returns `true`; otherwise, returns `false`).

Frames

- Frame
  - A graphical window on the screen.
- Component
  - Graphical widgets inside a frame, such as buttons or text input fields.

Frames


Icon
Title
Window-Buttons
java.awt.Frame
File Edit
Menu Bar (Optional)
Content Pane

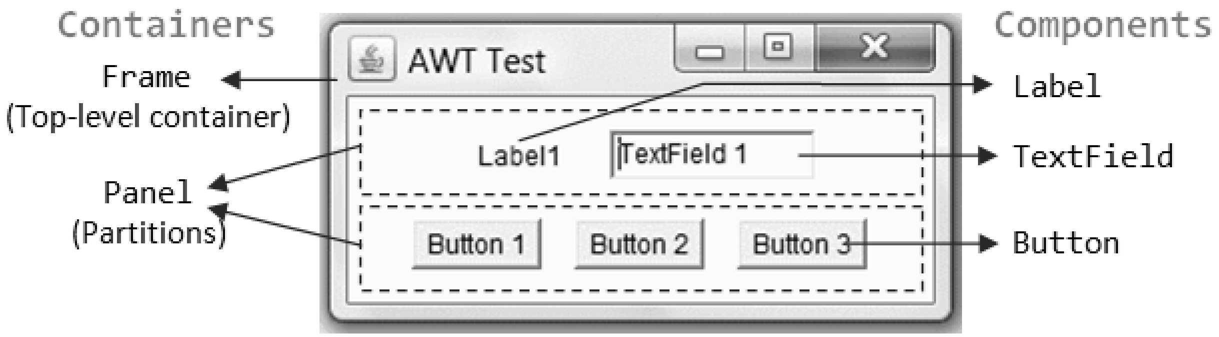

Containers
Frame (Top-level container)
Panel (Partitions)
AWT Test
Label1
TextField 1
Button 1
Button 2
Button 3
Components
Label
TextField
Button

Table 14.3 Useful Properties That Are Specific to `JFrames`

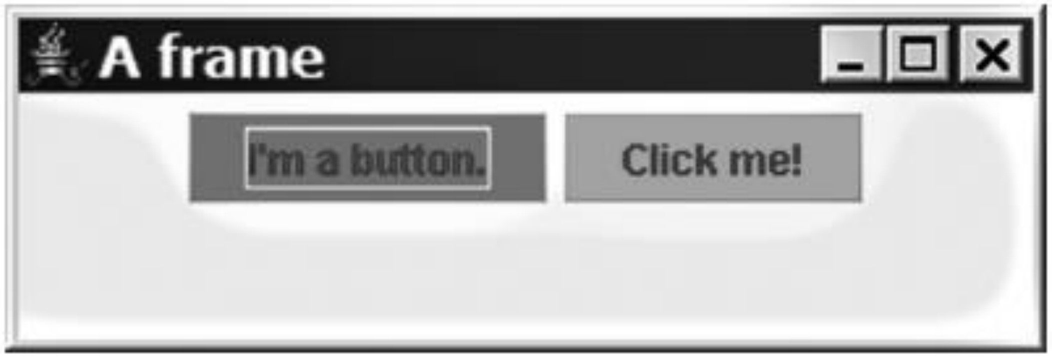
Property	Type	Description	Methods
default Close operation	`int`	What should happen when the frame is closed; choices include: • `JFrame.DO_NOTHING_ON_CLOSE` (don't do anything) • `JFrame.HIDE_ON_CLOSE` (hide the frame) • `JFrame.DISPOSE_ON_CLOSE` (hide and destroy the frame so that it cannot be shown again) • `JFrame.EXIT_ON_CLOSE` (exit the program)	`getDefaultCloseOperation`, `setDefaultCloseOperation(int)`
icon image	`Image`	The icon that appears in the title bar and Start menu or Dock	`getIconImage`, `setIconImage(Image)`
layout	`LayoutManager`	An object that controls the positions and sizes of the components inside this frame	`getLayout`, `setLayout(LayoutManager)`
resizable	`boolean`	Whether or not the frame allows itself to be resized	`isResizable`, `setResizable(boolean)`
title	`String`	The text that appears in the frame's title bar	`getTitle`, `setTitle(String)`

Table 14.4 Useful Properties of All Components (Including `JFrames`)

Property	Type	Description	Methods
background	`Color`	Background color	`getBackground,` `setBackground(Color)`
enabled	`boolean`	Whether the component can be interacted with	`isEnabled,` `setEnabled(boolean)`
focusable	`boolean`	Whether the keyboard can send input to the component	`isFocusable,` `setFocusable(boolean)`
font	`Font`	Font used to write text	`getFont, setFont(Font)`
foreground	`Color`	Foreground color	`getForeground,` `setForeground(Color)`
location	`Point`	(*x*, *y*) coordinate of component's top-left corner	`getLocation,` `setLocation(Point)`
size	`Dimension`	Current width and height of the component	`getSize,` `setSize(Dimension)`
preferred size	`Dimension`	"Preferred" width and height of the component; the size it should be to make it appear naturally on the screen (used with layout managers, seen later)	`getPreferredSize,` `setPreferredSize(Dimension)`
visible	`boolean`	Whether the component can be seen on the screen	`isVisible,` `setVisible(boolean)`

Layout Manager

- A Java object that determines the positions, sizes, and resizing behavior of the components within a frame or other container on the screen.
- **`import java.awt.*;`** // for layout managers
- **`import javax.swing.*;`** // for GUI components



Handling and Event

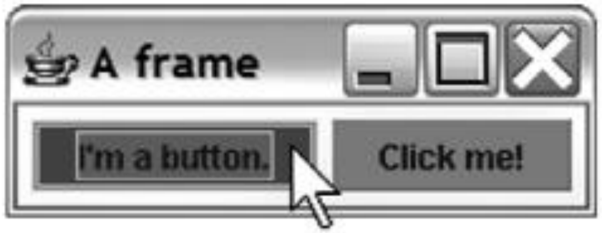
- **Event**
  - An object that represents a user's interaction with a GUI component and that can be handled by your programs to create interactive components.
- **Listener**
  - An object that is notified when an event occurs and that executes code to respond to the event.

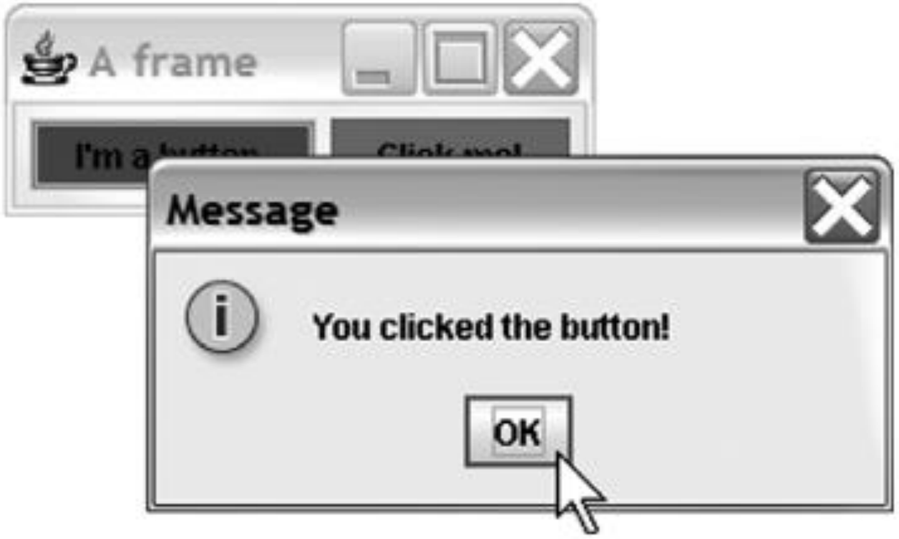
Handling and Event

- **Action Event**
  - An action event is a fairly general type of event that occurs when the user interacts with many standard components (for example, clicking on a button or entering text into a `JTextField`).
- **`ActionListener`**
  - The interface for handling action events in Java.

Handling and Event

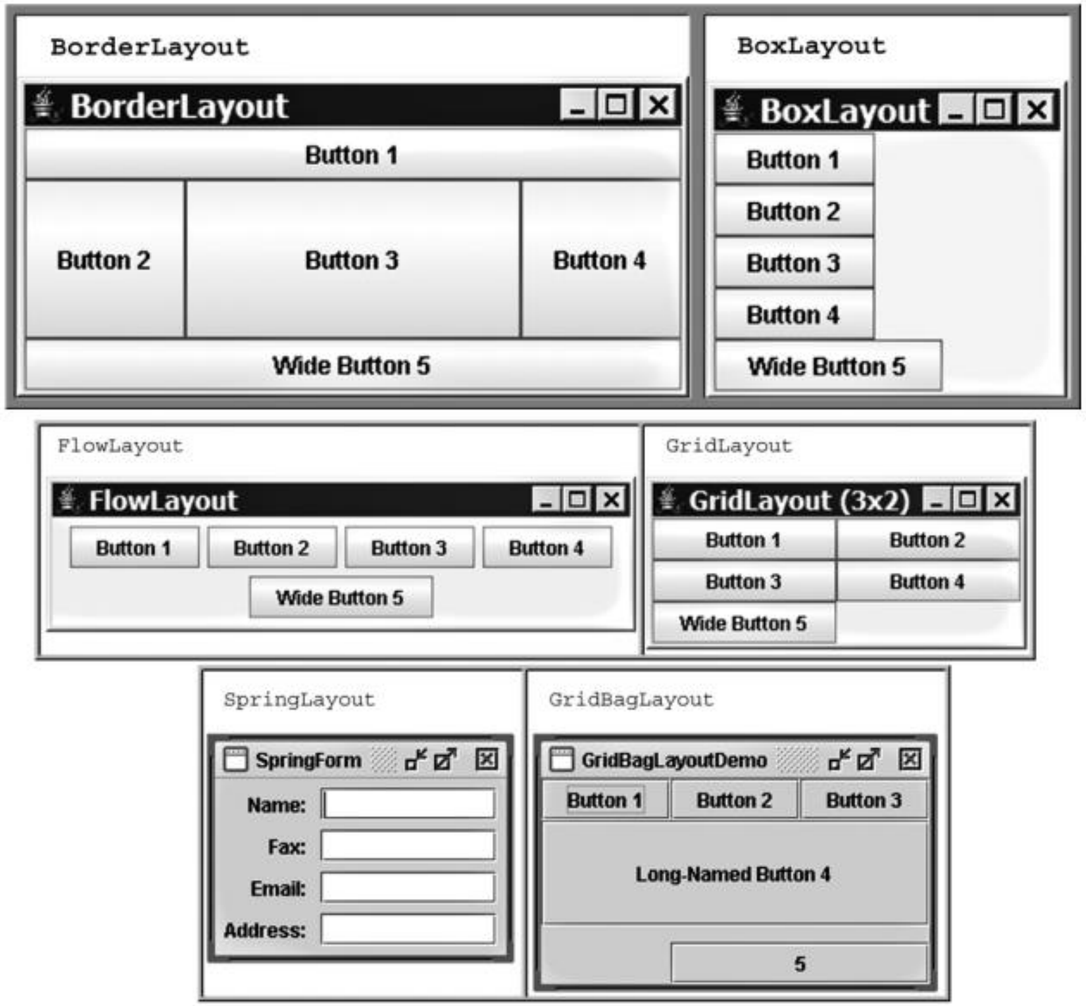
See example:





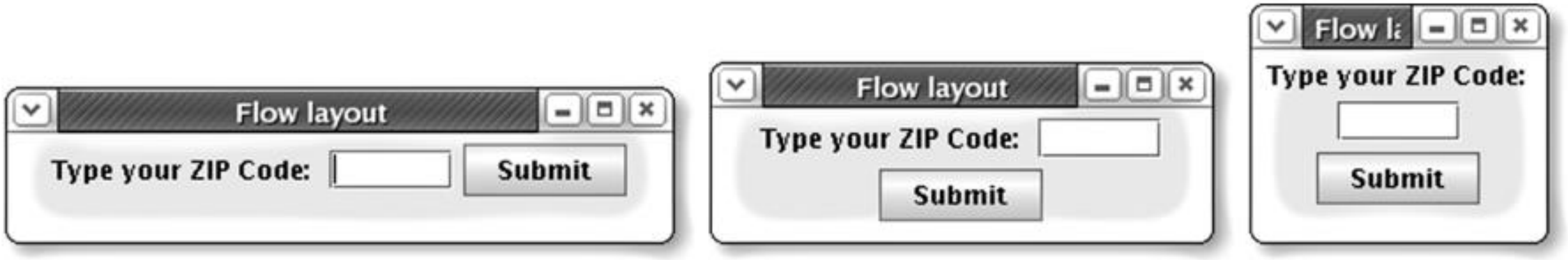
You clicked the button!

Components Layout

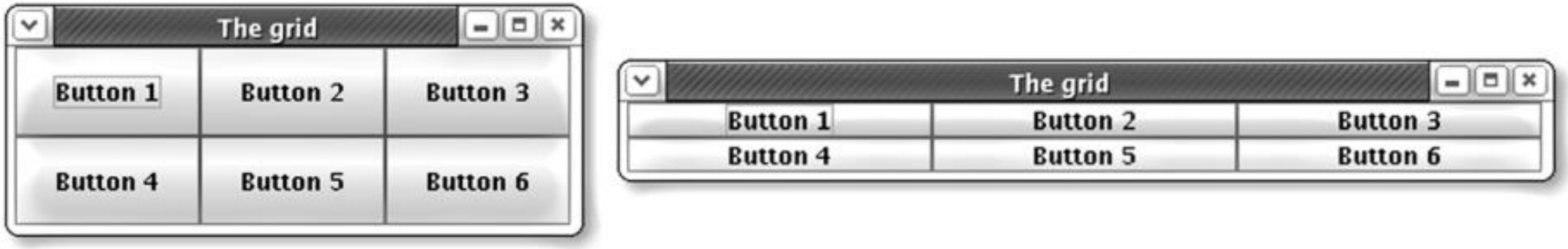


Components Layout

- Flow Layout

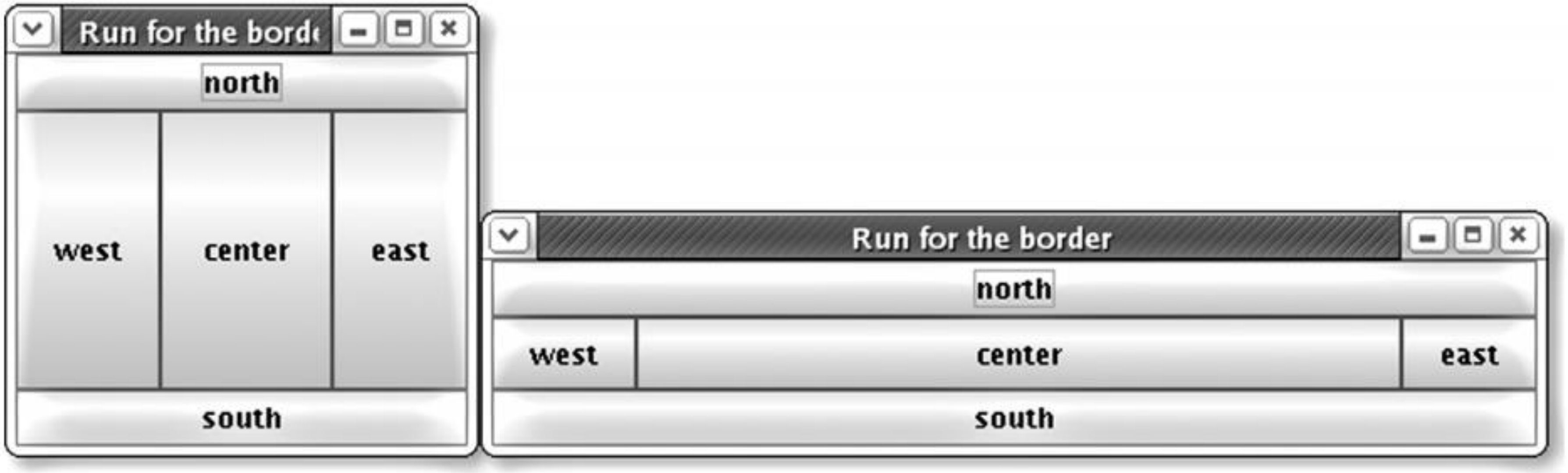


- Grid Layout



Components Layout

- Border Layout

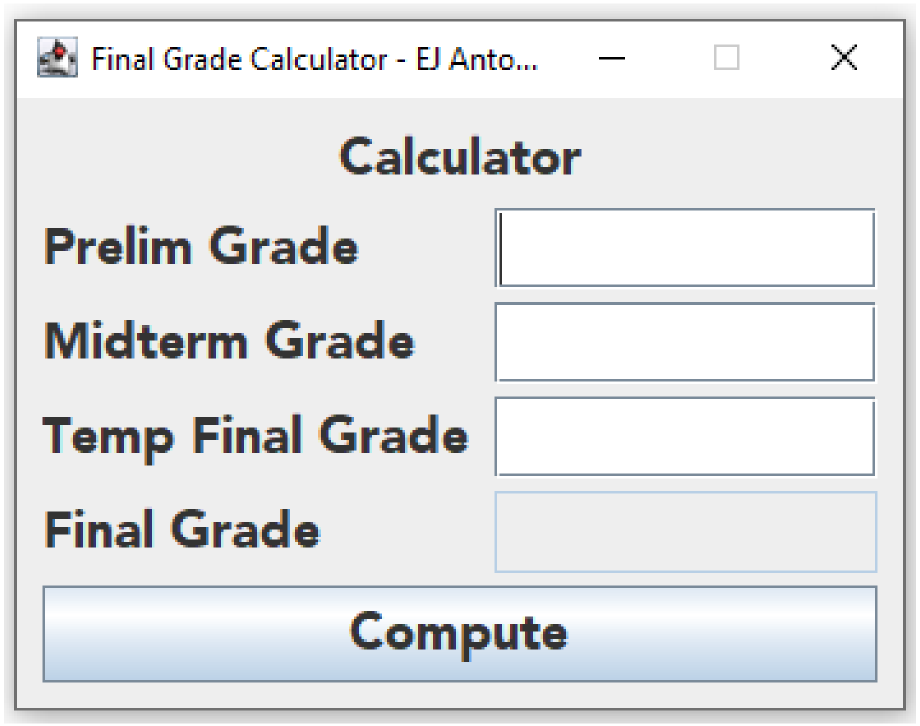


Components Layout

- **`FlowLayout:`**
  - Does not stretch components. Wraps to next line if necessary.
- **`GridLayout:`**
  Stretches all components in both dimensions to make them equal in size at all times.
- **`BorderLayout:`**
  - Stretches north and south regions horizontally but not vertically, stretches west and east regions vertically but not horizontally, and stretches center region in both dimensions to fill all remaining space not claimed by the other four regions.

Let's try!

- Create a GUI program that will compute for Final Grade. FG = (PG/9) + (2MG/9) + 2TFG / 3)



End of Module.

REFERENCE:

Programiz. (n.d.) .Learn Java Programming. https://www.programiz.com/java-programming

Singh, C. (n.d.). Beginners Book. https://beginnersbook.com/2017/08/java-break-statement/

Singh, C. (n.d.). Beginners Book https://beginnersbook.com/2017/08/java-continue-statement/