LECTURE 11: GUI APPLICATION

greenwich.edu.vn

UNIVERSITY of GREENWICH

Alliance with FPT Education

UNIVERSITY of GREENWICH
Alliance with FPT Education

PLAN

- GUI
- GUI Components
- Event Handler

UNIVERSITY of GREENWICH
Alliance with FPT Education

Introduction

- A graphical user interface (GUI) presents a user-friendly mechanism for interacting with an application.
  - Pronounced “GOO-ee”
  - Gives an application a distinctive “look” and “feel.”
  - Consistent, intuitive user-interface components give users a sense of familiarity
  - Learn new applications more quickly and use them more productively.

UNIVERSITY of GREENWICH
Alliance with FPT Education

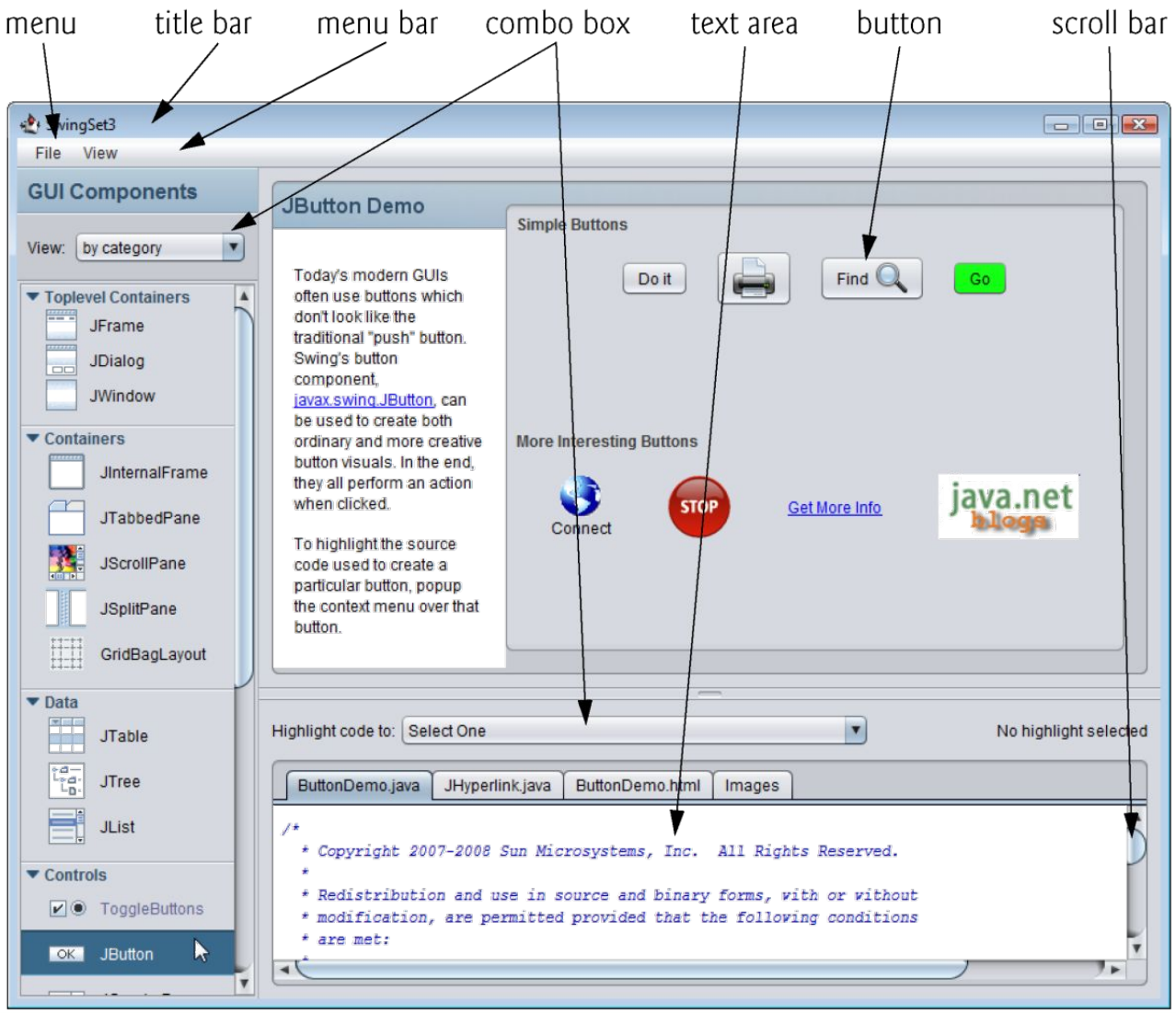
Introduction (cont.)

- Built from GUI components.
  - Sometimes called controls or widgets—short for window gadgets.
- User interacts via the mouse, the keyboard or another form of input, such as voice recognition.
- IDEs
  - Provide GUI design tools to specify a component's exact size and location in a visual manner by using the mouse.
  - Generates the GUI code for you.
  - Greatly simplifies creating GUIs, but each IDE has different capabilities and generates different code.

UNIVERSITY of GREENWICH

Alliance with FPT Education

GUI Example



UNIVERSITY of GREENWICH
Alliance with FPT Education

Overview of Swing Components

- Swing GUI components located in package javax.swing.
- Abstract Window Toolkit (AWT) in package java.awt is another set of GUI components in Java.
  - When a Java application with an AWT GUI executes on different Java platforms, the application's GUI components display differently on each platform.
- Together, the appearance and the way in which the user interacts with the application are known as that application's look-and-feel.
- Swing GUI components allow you to specify a uniform look-and-feel for your application across all platforms or to use each platform's custom look-and-feel.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Overview of Swing Components

- Most Swing components are not tied to actual GUI components of the underlying platform.
  - Known as lightweight components.
- AWT components are tied to the local platform and are called heavyweight components, because they rely on the local platform's windowing system to determine their functionality and their look-and-feel.
- Several Swing components are heavyweight components.

UNIVERSITY of GREENWICH
Alliance with FPT Education

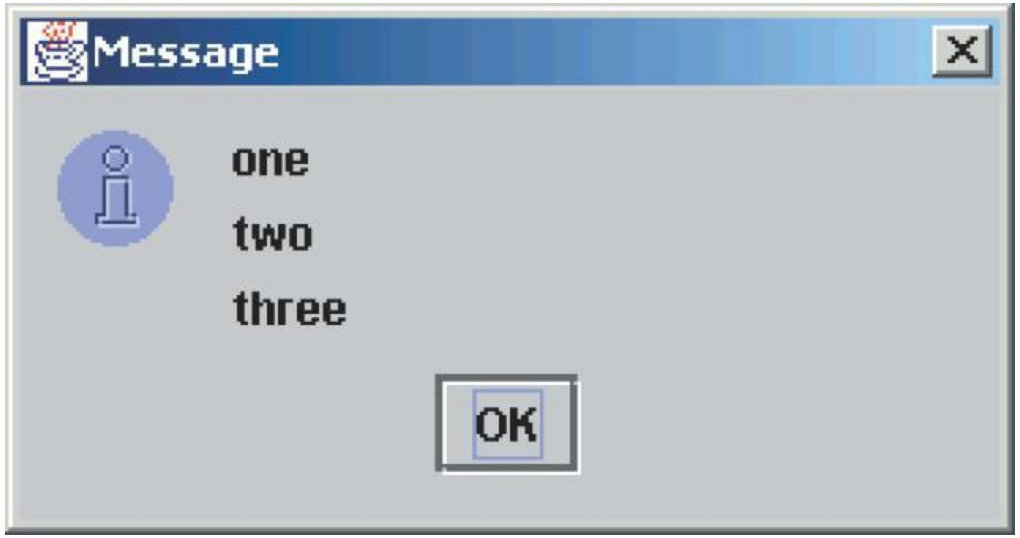
Using JOptionPane for Output

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

JOptionPane.showMessageDialog( null, "I Love Java" );
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Using JOptionPane for Output - 2

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

JOptionPane.showMessageDialog( null, "one\ntwo\nthree"
);
//place newline \n to display multiple lines of output
```



UNIVERSITY of GREENWICH
Alliance with FPT Education

JOptionPane for Input

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

String inputstr =

JOptionPane.showInputDialog( null, "What is your name?" );
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

JOptionPane static constants

Message dialog type	Icon	Description
`ERROR_MESSAGE`		Indicates an error.
`INFORMATION_MESSAGE`		Indicates an informational message.
`WARNING_MESSAGE`		Warns of a potential problem.
`QUESTION_MESSAGE`		Poses a question. This dialog normally requires a response, such as clicking a **Yes** or a **No** button.
`PLAIN_MESSAGE`	no icon	A dialog that contains a message, but no icon.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Displaying Text and Images in a Window

- Most windows that can contain Swing GUI components are instances of class JFrame or a subclass of JFrame.
- JFrame is an indirect subclass of class java.awt.Window
- Provides the basic attributes and behaviors of a window
  - a title bar at the top
  - buttons to minimize, maximize and close the window
- Most of our examples will consist of two classes
  - a subclass of JFrame that demonstrates new GUI concepts
  - an application class in which main creates and displays the application's primary window.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Subclassing JFrame

- To create a customized frame window, we define a subclass of the JFrame class.

- The JFrame class contains rudimentary functionalities to support features found in any frame window.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Creating a Plain JFrame

```
import javax.swing.*;
class DefaultJFrame {
    public static void main( String[] args ) {
        JFrame defaultJFrame;
        defaultJFrame = new JFrame();
        defaultJFrame.setVisible(true);
    }
}
```

You may not notice this frame window on the screen at first because it is so small. Look carefully at the top left corner of the screen.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Creating a Subclass of JFrame

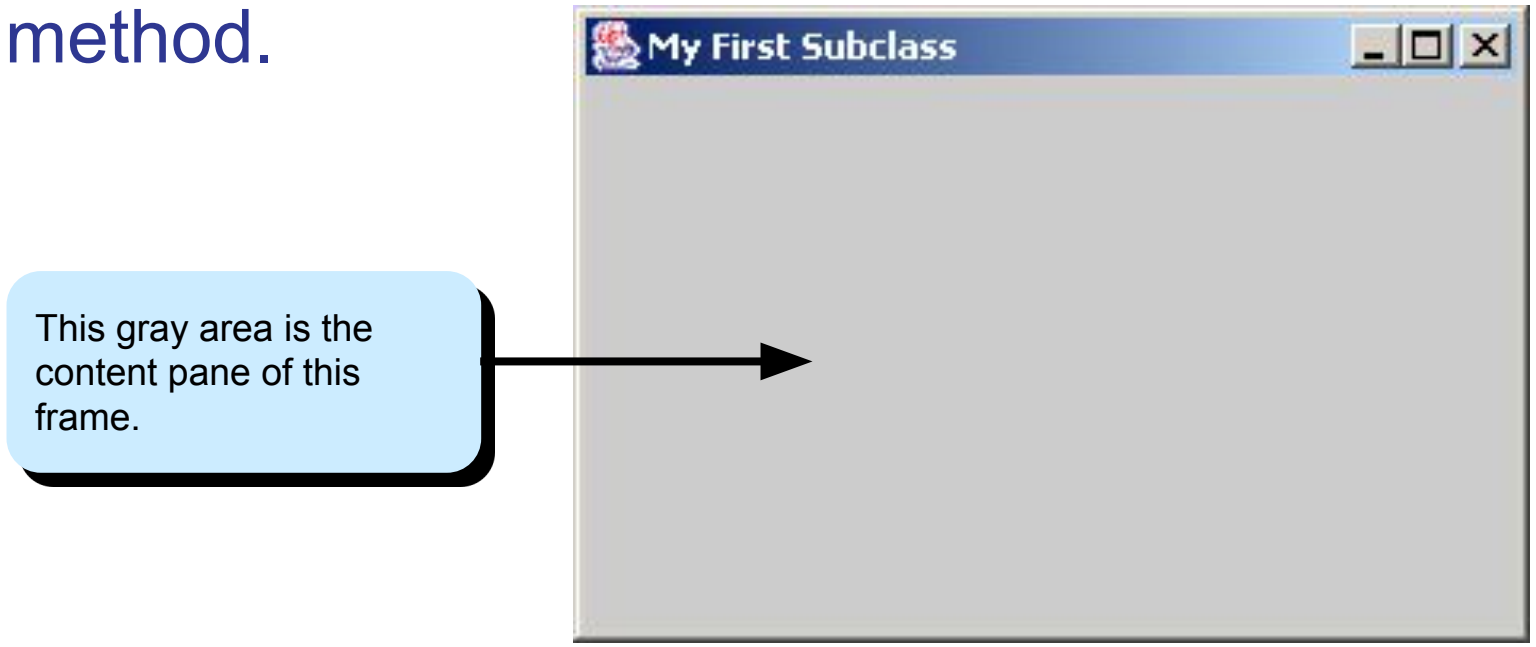
- To define a subclass of another class, we declare the subclass with the reserved word extends.

```
import javax.swing.*;

class Ch7JFrameSubclass1 extends JFrame {
    . . .
}
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

The Content Pane of a Frame

- The content pane is where we put GUI objects such as buttons, labels, scroll bars, and others.
- We access the content pane by calling the frame's getContentPane method.

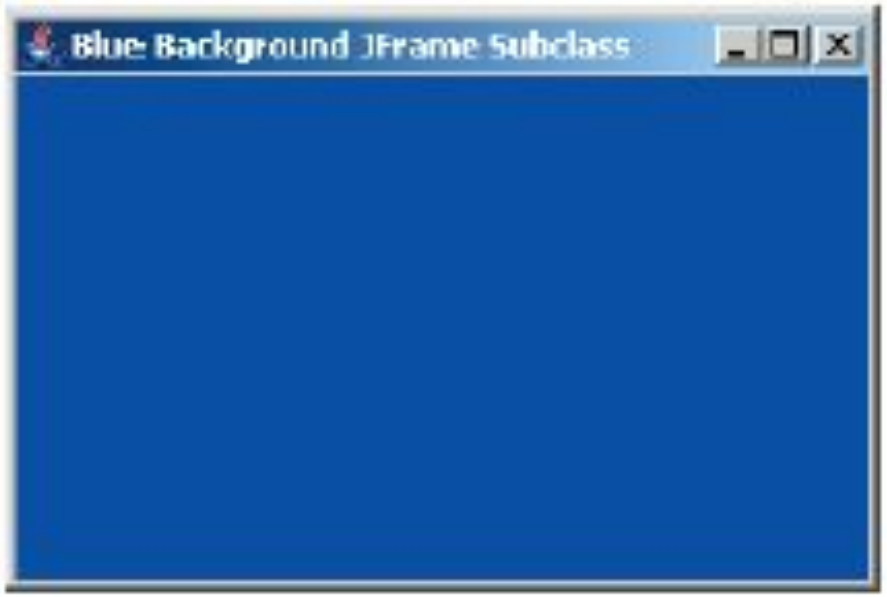


UNIVERSITY of GREENWICH
Alliance with FPT Education

Changing the Background Color

- Here's how we can change the background color of a content pane to blue:

```
Container contentPane = getContentPane();
contentPane.setBackground(Color.BLUE);
```



UNIVERSITY of GREENWICH
Alliance with FPT Education

Placing GUI Objects on a Frame

- There are two ways to put GUI objects on the content pane of a frame:
  - Use a *layout manager*
    - FlowLayout
    - BorderLayout
    - GridLayout
  - Use *absolute positioning*
    - null layout manager

UNIVERSITY of GREENWICH
Alliance with FPT Education

Placing a Button

- A JButton object a GUI component that represents a pushbutton.
- Here's an example of how we place a button with FlowLayout.

```
contentPane.setLayout( new FlowLayout());
okButton = new JButton("OK");
cancelButton = new JButton("CANCEL");
contentPane.add(okButton);
contentPane.add(cancelButton);
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

IDE Help for GUI Desgin

- Many IDEs provide GUI design tools in which you can specify the exact size and location of a component
- IDE generates the GUI code for you
- Greatly simplifies GUI creation
- To ensure that this book's examples can be used with any IDE, we did not use an IDE to create the GUI code
- We use Java's layout managers in our GUI examples

UNIVERSITY of GREENWICH
Alliance with FPT Education

Layout Managers

- The layout manager determines how the GUI components are added to the container (such as the content pane of a frame)
- Among the many different layout managers, the common ones are
  - FlowLayout
  - BorderLayout
  - GridLayout

UNIVERSITY of GREENWICH
Alliance with FPT Education

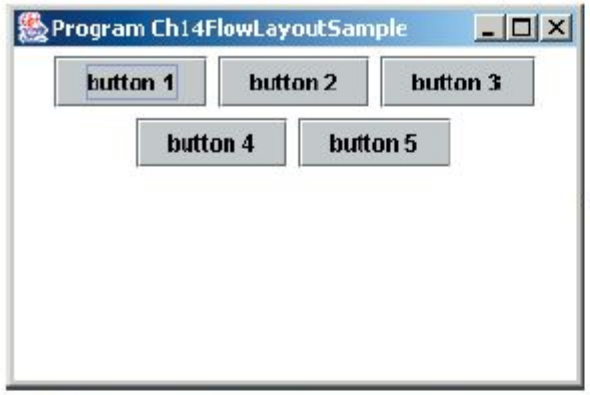
FlowLayout

- In using this layout, GUI componentsare placed in left-to-right order.
  - When the component does not fit on the same line, left-to-right placement continues on the next line.
- As a default, components on each line are centered.
- When the frame containing the component is resized, the placement of components is adjusted accordingly.

UNIVERSITY of GREENWICH
Alliance with FPT Education

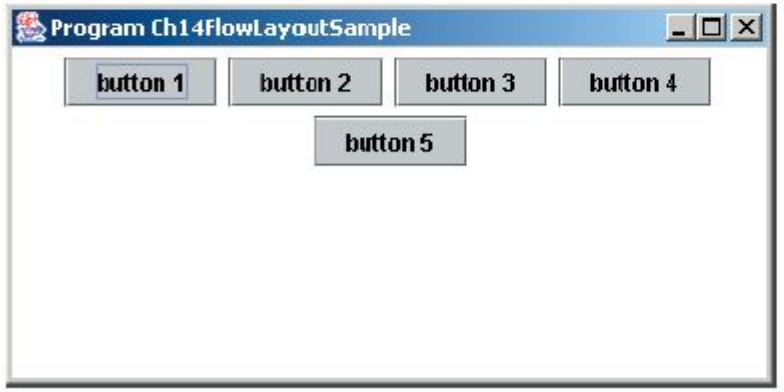
FlowLayout Sample

- This shows the placement of five buttons by using FlowLayout.



When the frame first appears on the screen.

Center alignment is used as a default. It can be set to a different align-ment at the time a FlowLayout is created.



After the frame's width is widened and shortened.

UNIVERSITY of GREENWICH
Alliance with FPT Education

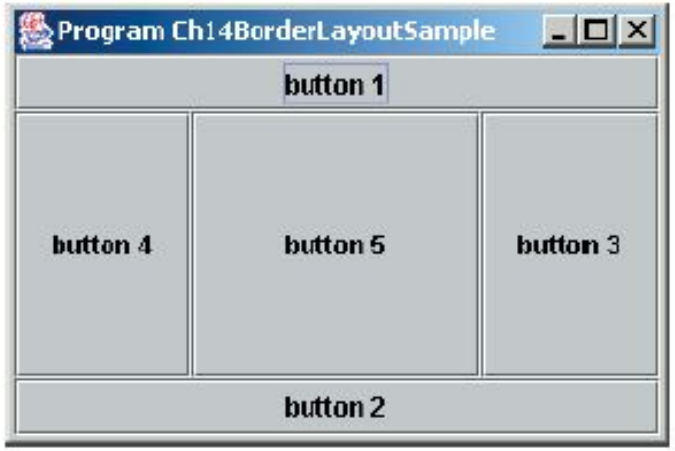
BorderLayout

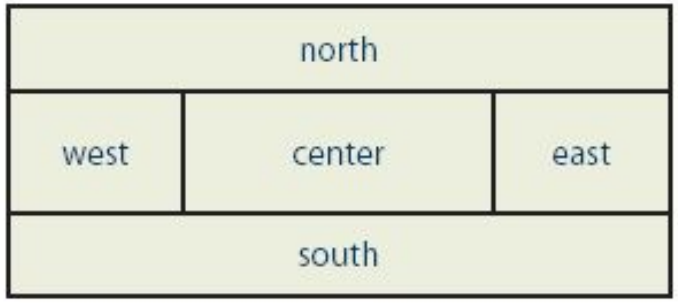
- This layout manager divides the container into five regions: center, north, south, east, and west.
- The north and south regions expand or shrink in height only
- The east and west regions expand or shrink in width only
- The center region expands or shrinks on both height and width.
- Not all regions have to be occupied.

UNIVERSITY of GREENWICH
Alliance with FPT Education

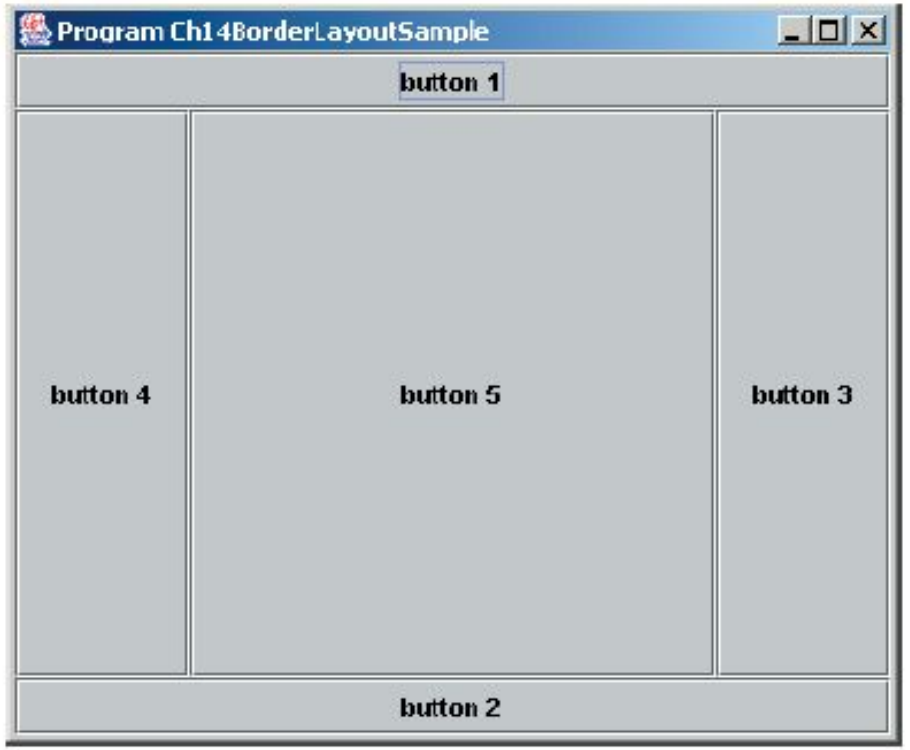
BorderLayout Sample

When the frame first appears on the screen.





After the frame is resized.



UNIVERSITY of GREENWICH
Alliance with FPT Education

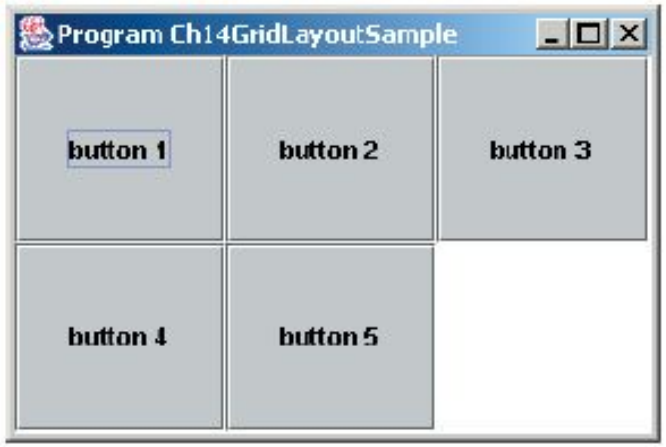
GridLayout

- This layout manager placesGUI components on equal-size N by M grids.
- Components are placed in top-to-bottom, left-to-right order.
- The number of rows and columns remains the same after the frame is resized, but the width and height of each region will change.

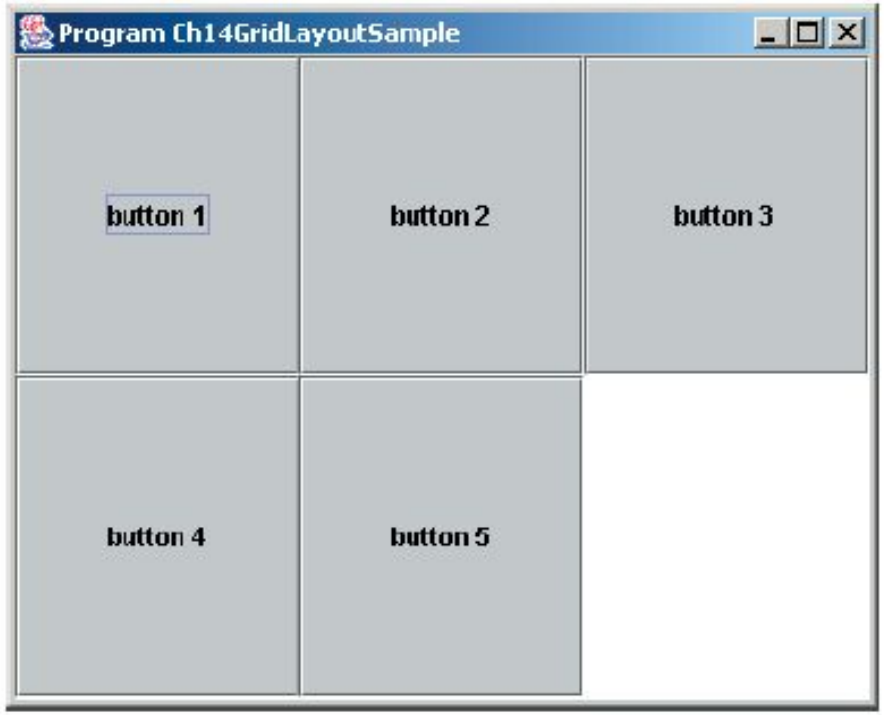
UNIVERSITY of GREENWICH
Alliance with FPT Education

GridLayout Sample

When the frame first appears on the screen.



After the frame is resized.



UNIVERSITY of GREENWICH
Alliance with FPT Education

Event Handling

- An action involving a GUI object, such as clicking a button, is called an event.
- The mechanism to process events is called event handling.
- The event-handling model of Java is based on the concept known as the delegation-based event model.
- With this model, event handling is implemented by two types of objects:
  - event source objects
  - event listener objects

UNIVERSITY of GREENWICH
Alliance with FPT Education

Event Source Objects

- An event source is a GUI object where an event occurs. We say an event source generates events.
- Buttons, text boxes, list boxes, and menus are common event sources in GUI-based applications.
- Although possible, we do not, under normal circumstances, define our own event sources when writing GUI-based applications.

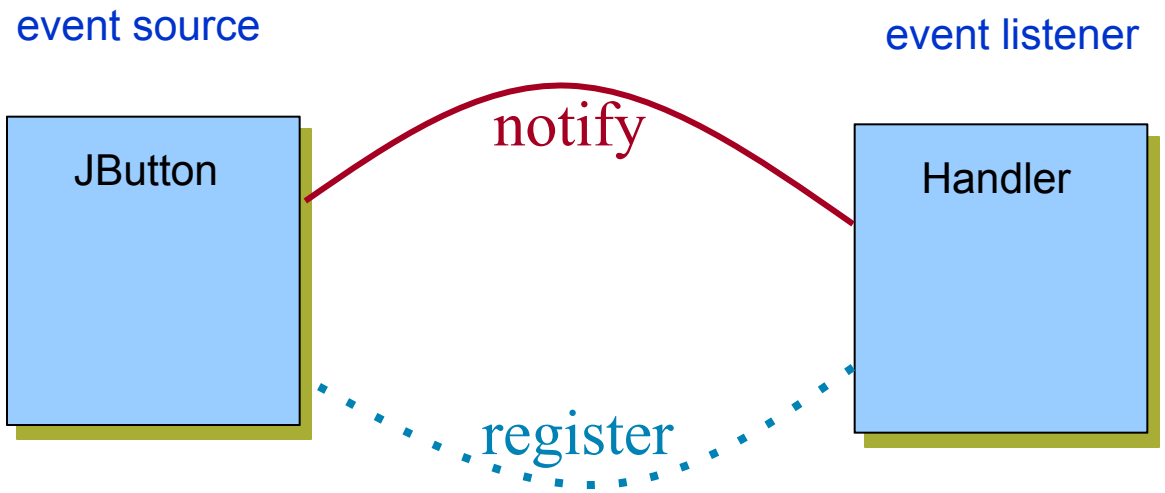
UNIVERSITY of GREENWICH
Alliance with FPT Education

Event Listener Objects

- An event listener object is an object that includes a method that gets executed in response to the generated events.
- A listener must be associated, or registered, to a source, so it can be notified when the source generates events.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Connecting Source and Listener



A listener must be registered to a event source. Once registered, it will get notified when the event source generates events.

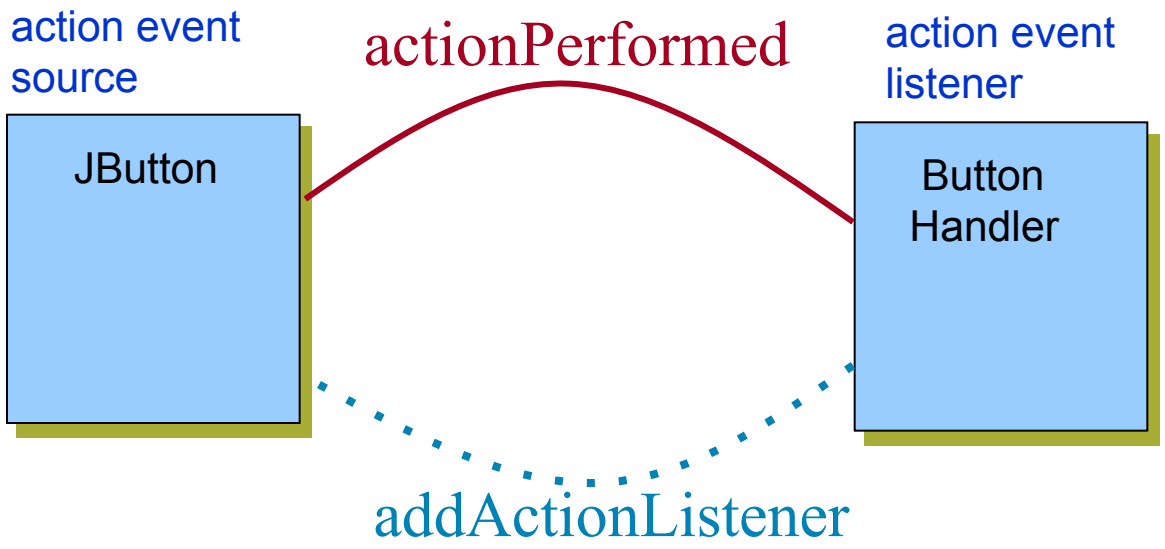
UNIVERSITY of GREENWICH
Alliance with FPT Education

Event Types

- Registration and notification are specific to event types
  - Mouse listener handles mouse events
  - Item listener handles item selection events
  - and so forth
- Among the different types of events, the action event is the most common.
  - Clicking on a button generates an action event
  - Selecting a menu item generates an action event
  - and so forth
- Action events are generated by action event sources and handled by action event listeners.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Handling Action Events




```
JButton button = new JButton("OK");
ButtonHandler handler = new ButtonHandler( );

button.addActionListener(handler);
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

The Java Interface

- A Java interface includes only constants and abstract methods.
- An abstract method has only the method header, or prototype. There is no method body. You cannot create an instance of a Java interface.
- A Java interface specifies a behavior.
- A class implements an interface by providing the method body to the abstract methods stated in the interface.
- Any class can implement the interface.

UNIVERSITY of GREENWICH
Alliance with FPT Education

ActionListener Interface

- When we call the addActionListener method of an event source, we must pass an instance of a class that implements the ActionListener interface.
- The ActionListener interface includes one method named actionPerformed.
- A class that implements the ActionListener interface must therefore provide the method body of actionPerformed.
- Since actionPerformed is the method that will be called when an action event is generated, this is the place where we put a code we want to be executed in response to the generated events.

UNIVERSITY of GREENWICH
Alliance with FPT Education

The ButtonHandler Class

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

class ButtonHandler implements ActionListener {
    . . .
    public void actionPerformed(ActionEvent event) {
        JButton clickedButton = (JButton) event.getSource();

        JRootPane rootPane = clickedButton.getRootPane( );
        Frame     frame    = (JFrame) rootPane.getParent();

        frame.setTitle("You clicked " + clickedButton.getText());
    }
}
```

Container as Event Listener

- Instead of defining a separate event listener such as ButtonHandler, it is much more common to have an object that contains the event sources be a listener.
  - Example: We make this frame a listener of the action events of the buttons it contains.

UNIVERSITY of GREENWICH
Alliance with FPT Education

ButtonFrameHandlerDemo

```
. . .
class JButtonFrameHandlerDemo extends JFrame implements ActionListener {
    . . .
    public void actionPerformed(ActionEvent event) {
        JButton clickedButton    = (JButton) event.getSource();

        String  buttonText = clickedButton.getText();

        setTitle("You clicked " + buttonText);
    }
}
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

GUI Classes for Handling Text

- The Swing GUI classes JLabel, JTextField, and JTextArea deal with text.
- A JLabel object displays uneditable text (or image).
- A JTextField object allows the user to enter a single line of text.
- A JTextArea object allows the user to enter multiple lines of text. It can also be used for displaying multiple lines of uneditable text.

UNIVERSITY of GREENWICH
Alliance with FPT Education

JTextField

- We use a JTextField object to accept a single line to text from a user. An action event is generated when the user presses the ENTER key.
- The getText method of JTextField is used to retrieve the text that the user entered.

```
JTextField input = new JTextField( );
input.addActionListener(eventListener);
contentPane.add(input);
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

JLabel

- We use a JLabel object to display a label.
- A label can be a text or an image.
- When creating an image label, we pass ImageIcon object instead of a string.

```
JLabel textLabel = new JLabel("Please enter your name");
contentPane.add(textLabel);

JLabel imgLabel = new JLabel(new ImageIcon("cat.gif"));
contentPane.add(imgLabel);
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Ch14TextFrame2

JLabel
(with an image)

JLabel
(with a text)

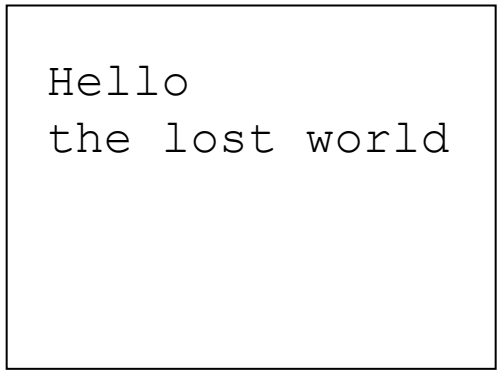
JTextField

UNIVERSITY of GREENWICH
Alliance with FPT Education

JTextArea

- We use a JTextArea object to display or allow the user to enter multiple lines of text.
- The setText method assigns the text to a JTextArea, replacing the current content.
- The append method appends the text to the current text.

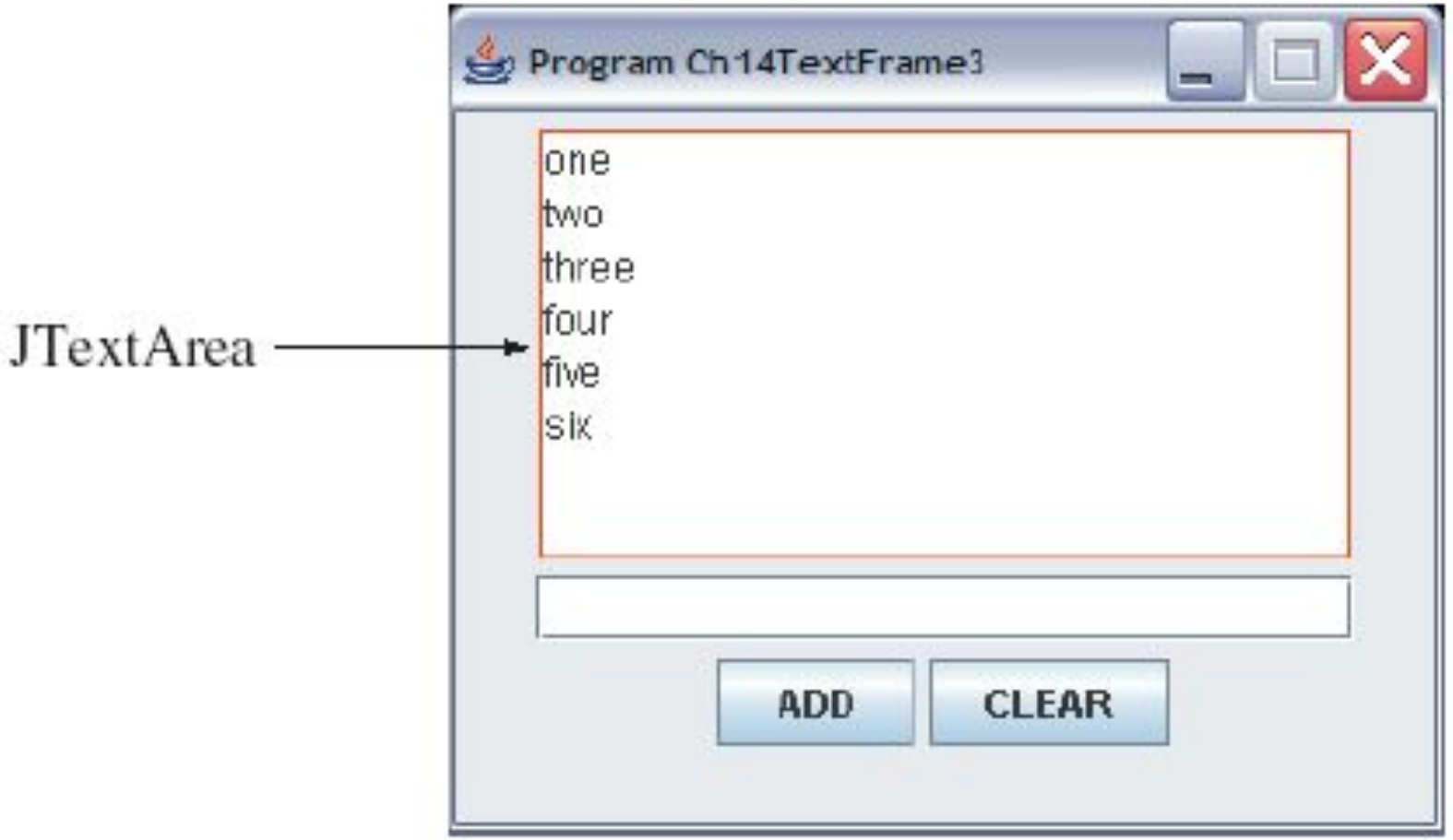
```
JTextArea textArea
        = new JTextArea( );
. . .
textArea.setText("Hello\n");
textArea.append("the lost ");
textArea.append("world");
```



JTextArea

UNIVERSITY of GREENWICH
Alliance with FPT Education

Demo



UNIVERSITY of GREENWICH
Alliance with FPT Education

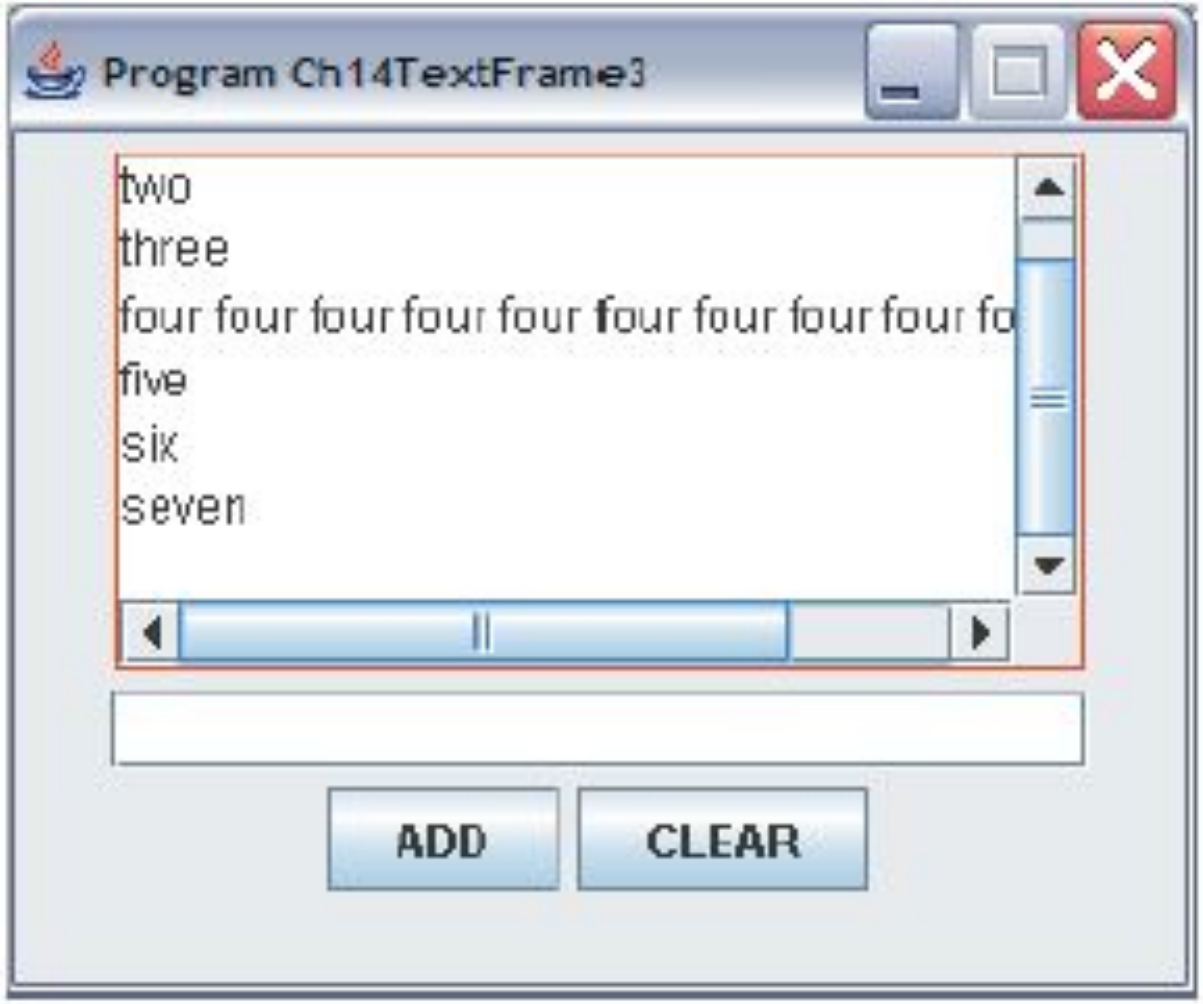
Adding Scroll Bars to JTextArea

- By default a JTextArea does not have any scroll bars. To add scroll bars, we place a JTextArea in a JScrollPane object.

```
JTextArea   textArea   = new JTextArea();
. . .
JScrollPane scrollText = new JScrollPane(textArea);
. . .
contentPane.add(scrollText);
```

UNIVERSITY of GREENWICH
Alliance with FPT Education

Demo



UNIVERSITY of GREENWICH
Alliance with FPT Education

Other Common GUI Components

- JCheckBox
- JRadioButton
- JComboBox
- JList
- JSlider

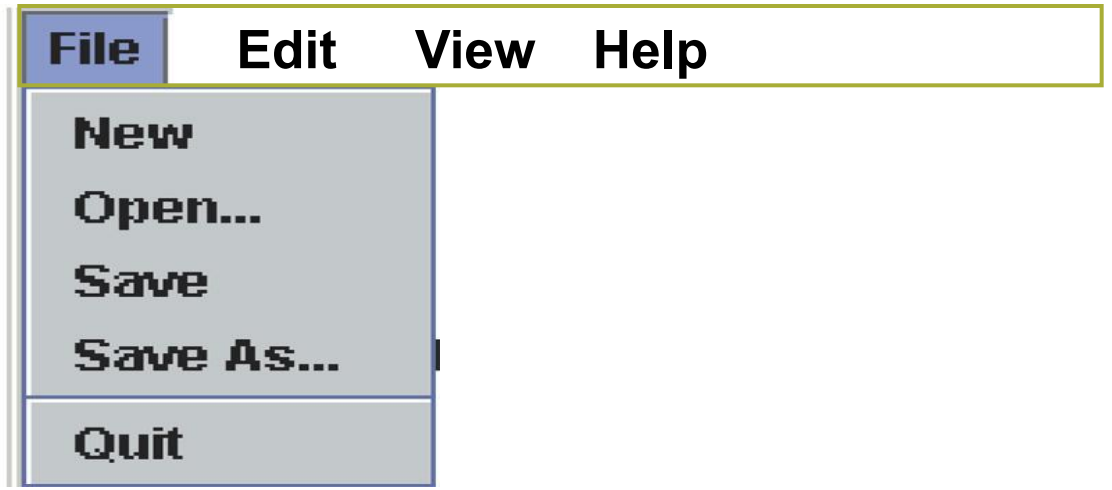
UNIVERSITY of GREENWICH
Alliance with FPT Education

Menus

- The javax.swing package contains three menu-related classes: JMenuBar, JMenu, and JMenuItem.
- JMenuBar is a bar where the menus are placed. There is one menu bar per frame.
- JMenu (such as File or Edit) is a group of menu choices. JMenuBar may include many JMenu objects.
- JMenuItem (such as Copy, Cut, or Paste) is an individual menu choice in a JMenu object.
- Only the JMenuItem objects generate events.

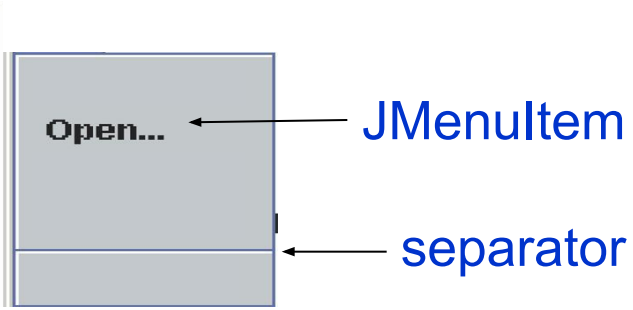
UNIVERSITY of GREENWICH
Alliance with FPT Education

Menu Components



JMenuBar

JMenu



UNIVERSITY of GREENWICH
Alliance with FPT Education

Sequence for Creating Menus

- Create a JMenuBar object and attach it to a frame.
- Create a JMenu object.
- Create JMenuItem objects and add them to the JMenu object.
- Attach the JMenu object to the JMenuBar object.

UNIVERSITY of GREENWICH
Alliance with FPT Education

Demo

- Demo a GUI application with several components working together, loading data from file to manage