



Lecture # 5

Imran Ahsan

Outline

- Graphical User Interface (GUI)
- AWT (Abstract Windowing Toolkit)
- Component & Containers
- AWT Packages
- AWT Container Classes
- AWT Component Classes
- Swing
- Swing Container Classes
- Swing Component Classes

Graphical User Interface (GUI)

- A graphical user interface (**GUI**) presents a **user-friendly mechanism** for **interacting** with an application.
- A GUI gives an application a **distinctive “look” and “feel.”**
- Providing different applications **with consistent, intuitive user interface** components allows users to be somewhat familiar with an application, so that they can learn it more quickly and use it more productively.
- A **GUI component is an object** with which the user interacts via the mouse, the keyboard or another form of input, such as voice recognition.

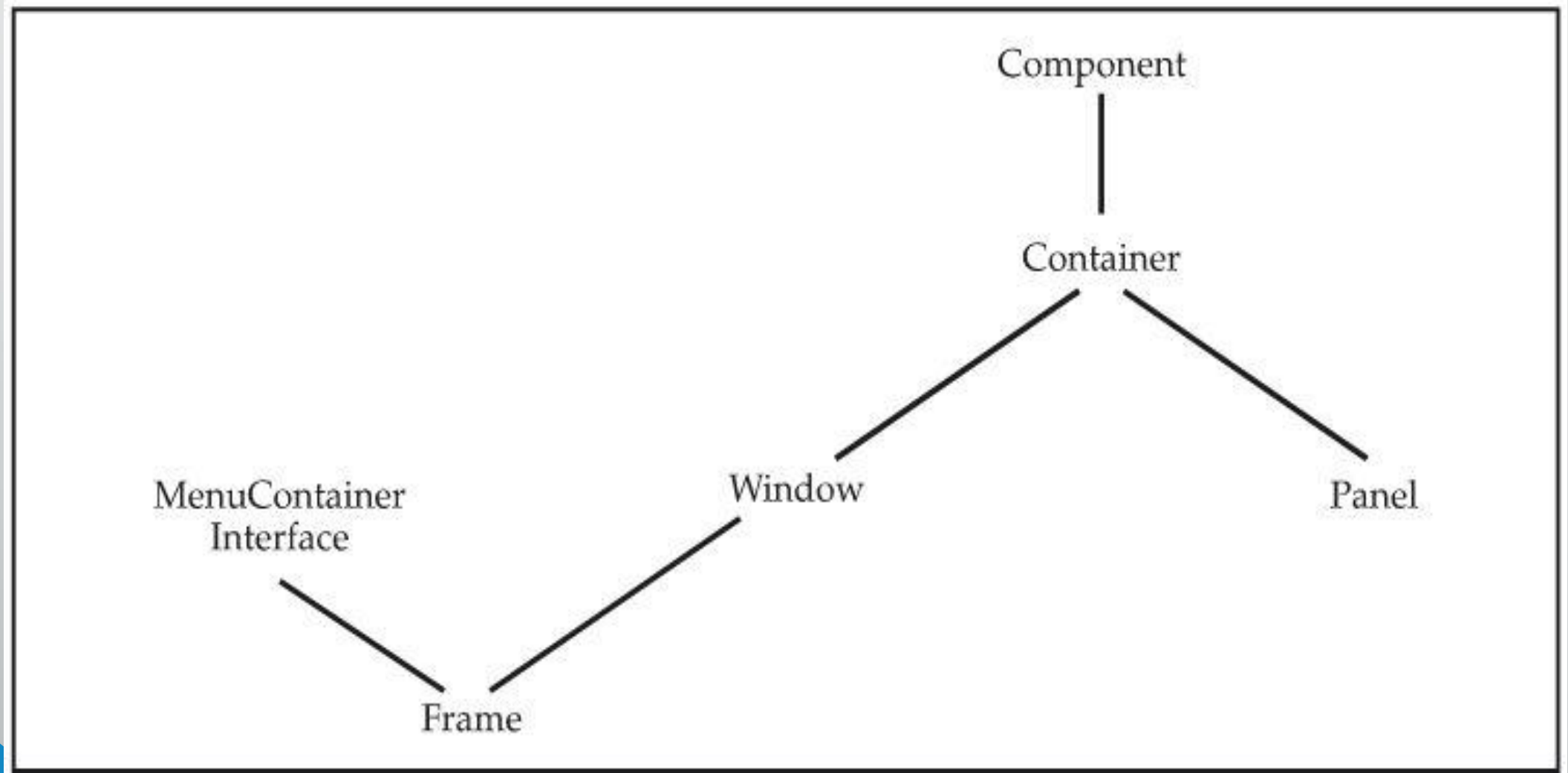
Graphical User Interface (GUI)

- There are **two** sets of Java APIs for graphics programming:
- AWT (Abstract Windowing Toolkit)
- Swing.

AWT(Abstract Windowing Toolkit)

- The AWT contains numerous classes and methods that allow you to create and manage windows.
- The AWT defines windows according to a class hierarchy that adds functionality and specificity with each level.
- The two most common windows are those derived from Panel, which is used by applets, and those derived from Frame, which creates a standard window.
- Much of the functionality of these windows is derived from their parent classes.
- AWT provides a *platform-independent* and *device-independent* interface to develop graphic programs that runs on all platforms, including Windows, Mac OS, and Unixes.

AWT(Abstract Windowing Toolkit)



AWT(Abstract Windowing Toolkit)

Component:

- At the **top** of the AWT hierarchy is the Component class.
- Component is an **abstract class** that encapsulates all of the attributes of a visual component.
- All **user interface elements** that are displayed on the screen and that interact with the user are **subclasses of Component**.
- It **defines over a hundred public methods** that are **responsible for managing events**, such as mouse and keyboard input, positioning and sizing the window, and repainting.
- A Component object is **responsible for remembering** the current foreground and background colors and the currently selected text font.

AWT(Abstract Windowing Toolkit)

Container:

- The Container class is a **subclass of Component**.
- It has **additional methods** that allow other Component objects to be nested within it.
- Other Container **objects can be stored inside of a Container** (since they are themselves instances of Component).
- This makes for a **multileveled containment system**.
- A container is **responsible for laying out** (that is, positioning) any components that it contains. It does this through the use of various layout managers.

AWT(Abstract Windowing Toolkit)

Panel:

- The Panel class is a **concrete subclass of Container**.
- It **doesn't add any new methods**; it simply **implements Container**.
- Panel is the **superclass for Applet** .
- A Panel is a window that **does not contain a title bar, menu bar, or border**. This is why you don't see these items when an applet is run inside a browser. When you run an applet using an applet viewer, the applet viewer provides the title and border.
- Other components can be added to a Panel object by its add() method (inherited from Container). Once these components have been added, you can position and resize them manually using the setLocation(), setSize(), or setBounds() methods defined by Component.

AWT(Abstract Windowing Toolkit)

Window:

- The Window class creates a **top-level window**.
- A top-level window is not contained within any other object; **it sits directly on the desktop**. Generally, you **won't create Window objects directly**. Instead, you will **use a subclass of Window called Frame**, described next.

Frame:

- Frame encapsulates what is commonly thought of as a **"window."**
- It is a **subclass of Window** and has a title bar, menu bar, borders, and resizing corners.
- When a Frame window is created by a program rather than an applet, a normal window is created.

AWT Packages

The `java.awt` package contains the core AWT graphics classes:

- GUI Component classes, such as Button, TextField, and Label,
- GUI Container classes, such as Frame and Panel,
- Layout managers, such as FlowLayout, BorderLayout and GridLayout,
- Custom graphics classes, such as Graphics, Color and Font.

AWT Container Classes

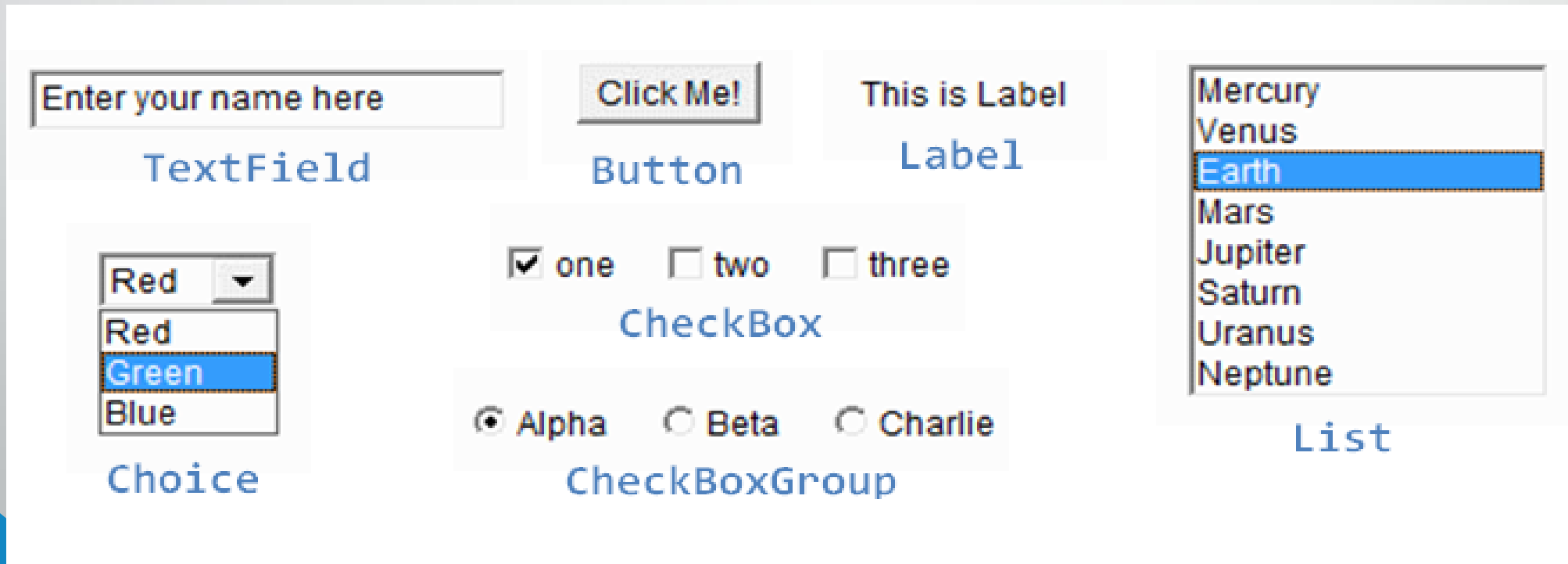
java.awt.*

- Frame, Panel, Window, Dialog and Applet
- Frame: **java.awt.Frame**
 - To write a GUI program, we typically start with a subclass extending from java.awt.Frame to inherit the main window.
- Panel: **java.awt.Panel**
 - a rectangular box under a higher-level container, used to layout a set of related GUI components in pattern such as grid or flow.
- Dialog: **java.awt.Dialog**
 - An AWT Dialog is a "pop-up window" used for interacting with the users. A Dialog has a title-bar (containing an icon, a title and a close button) and a content display area.
- Applet: **java.applet**
 - An AWT Applet (in package java.applet) is the top-level container for an applet, which is a Java program running inside a browser.

AWT Component Classes

(java.awt.*)

- AWT provides many ready-made and reusable GUI components in package java.awt. The frequently-used are: Button, TextField, Label, Checkbox, CheckboxGroup (radio buttons), List, and Choice, as illustrated below.



AWT Component Classes

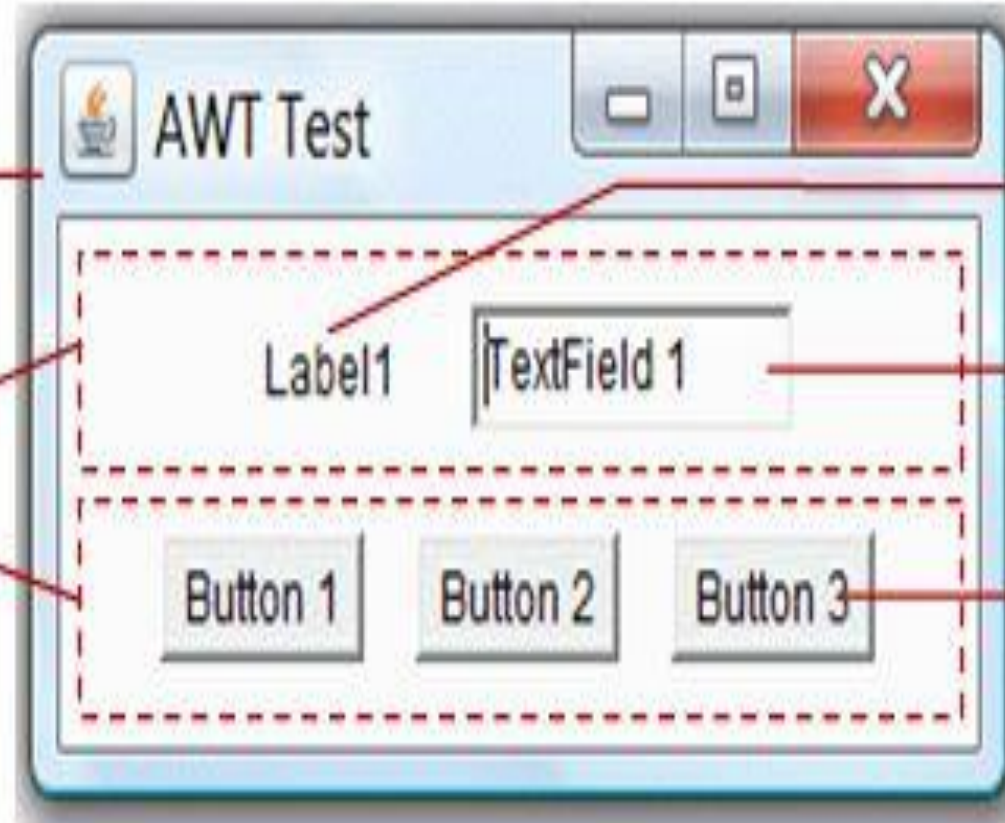
java.awt.*

- Frame, Panel, Window, Dialog and Applet
- Button: java.awt.Button
- Label: java.awt.Label
- Textfield: java.awt.TextField
- Checkbox: java.awt.Checkbox
- Etc.

Containers

Frame
(Top-level container)

Panel
(Partitions)



Components

Label

TextField

Button

- In the figure in previous slide, there are **three containers: a Frame and two Panels**. A Frame is the top-level container of an AWT program.
- In the figure, the top-level **Frame contains two Panels**. There are five components: a Label (providing description), a TextField (for users to enter text), and three Buttons (for user to trigger certain programmed actions).
- In a GUI program, **a component must be kept in a container**. You need to identify a container to hold the components. Every container has a method called `add(Component c)`. A container (say `c`) can invoke `c.add(aComponent)` to add `aComponent` into itself. For example:

- `Panel pnl = new Panel();` `// Panel is a container`
- `Button btn = new Button("Press");` `// Button is a component`
- `pnl.add(btn);` `// The Panel container adds a Button component`

Swing

- Swing is part of the "Java Foundation Classes (JFC)"
- Swing API: for advanced graphical programming.
- Swing is a rich set of easy-to-use, easy-to-understand JavaBean GUI components that can be dragged and dropped as "GUI builders" in visual programming environment.
- Compared with AWT, Swing provides a huge and comprehensive collection of reusable GUI components
- Swing components are *lightweight*. The AWT components are *heavyweight* (in terms of system resource utilization).

Swing Container Classes

`javax.swing.*`

- Frame, Panel, Window, Dialog and Applet
- Frame: `javax.swing.JFrame`
- Panel: `javax.swing.JPanel`
- Dialog: `javax.swing.JDialog`
- Applet: `javax.swing.JApplet`

Swing Component Classes

(`javax.swing.*`)

- Swing provides many ready-made and reusable GUI components in package `javax.swing`. The frequently-used are: `JButton`, `JTextField`, `JLabel`, `JCheckbox`, `JCheckboxGroup` (radio buttons), `JList`, and `JChoice`, as illustrated below.



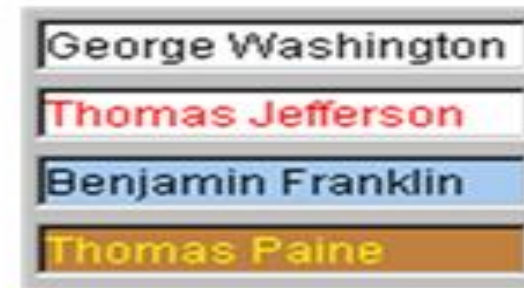
Buttons



Combo Box



List



TextField



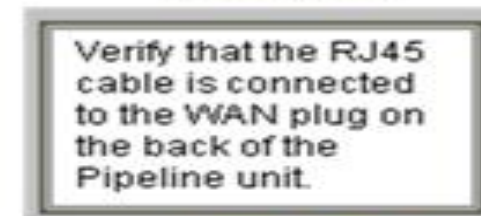
Slider



Menu



Label



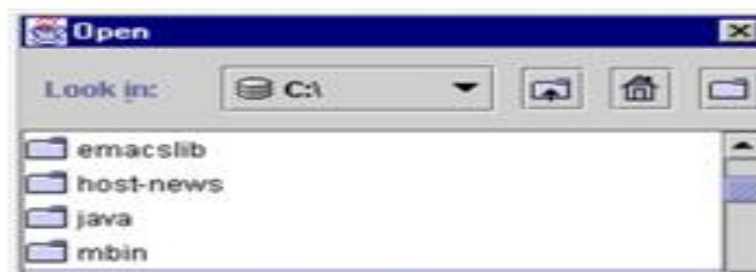
Text Area



Tool Tip



Progress Bar



File Chooser



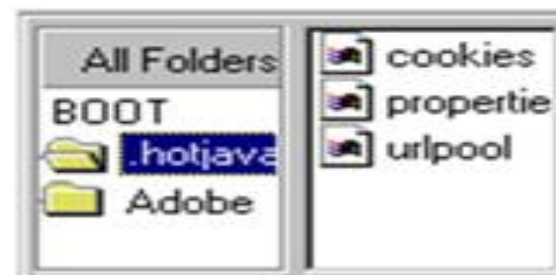
Color Chooser

First Na...	Last Name
Mark	Andrews
Tom	Ball
Alan	Chung
Jeff	Dinkins

Table



Tree



Split Pane



Tabbed Pane

Swing Component Classes

java.swing.*

Component	Description
JLabel	Displays uneditable text or icons.
TextField	Enables user to enter data from the keyboard. Can also be used to display editable or uneditable text.
Button	Triggers an event when clicked with the mouse.
CheckBox	Specifies an option that can be selected or not selected.
ComboBox	Provides a drop-down list of items from which the user can make a selection by clicking an item or possibly by typing into the box.
List	Provides a list of items from which the user can make a selection by clicking on any item in the list. Multiple elements can be selected.
Panel	Provides an area in which components can be placed and organized. Can also be used as a drawing area for graphics.



End.