# Java Swing

# AWT to Swing

- AWT: Abstract Windowing Toolkit
  - import java.awt.*
- Swing: new with Java2
  - import javax.swing.*
  - Extends AWT
  - Tons o‟ new improved components
  - Standard dialog boxes, tooltips, …
  - Look-and-feel, skins
  - Event listeners

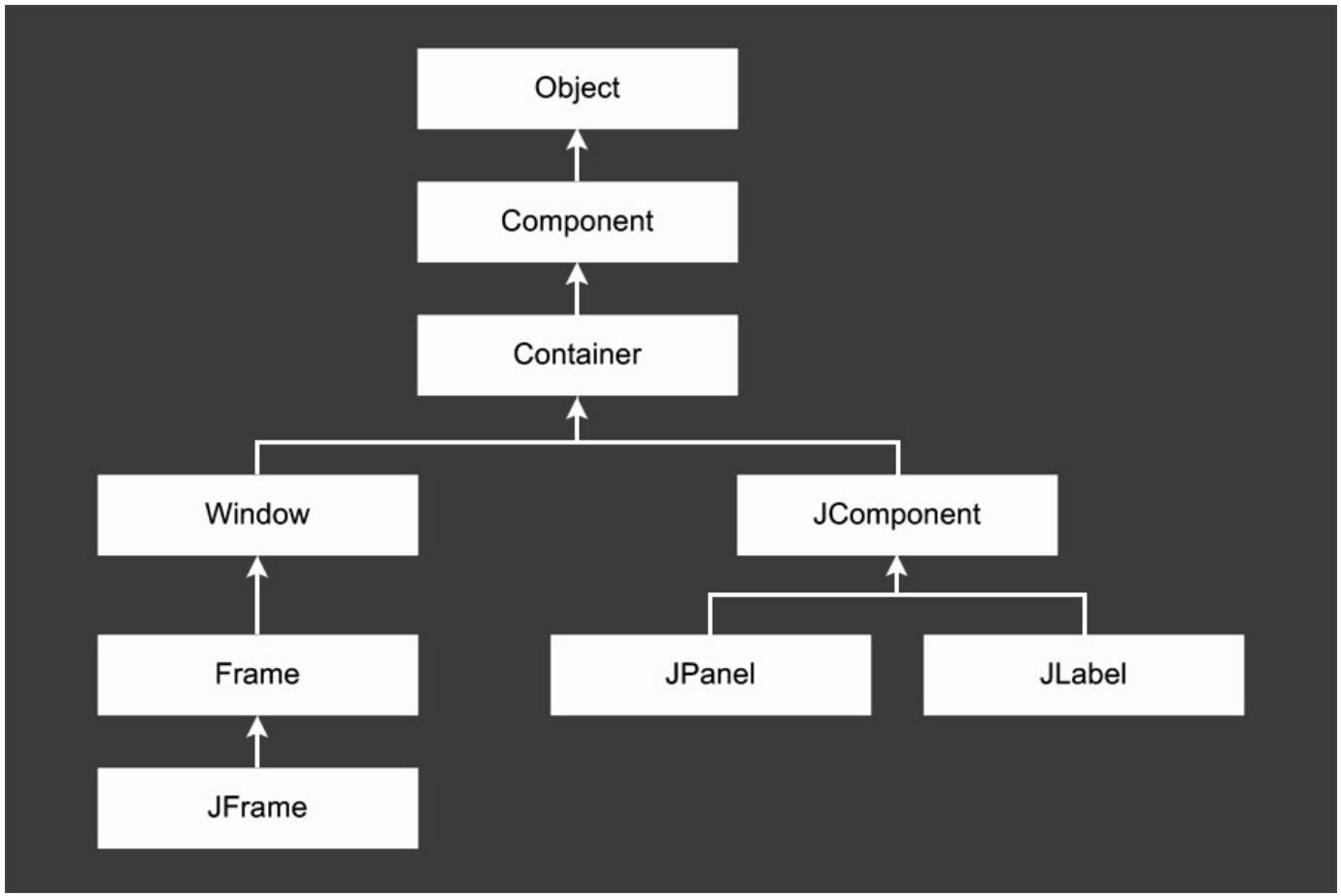
# What is Java Swing?

- Part of the Java Foundation Classes (JFC)
- Provides a rich set of GUI components
- Used to create a Java program with a graphical user interface (GUI)
- table controls, list controls, tree controls, buttons, and labels, and so on…

# What features are available?

- GUI components like button, checkbox, and so on…
- Java 2D API: images, figures, animation
- Pluggable look and feel: use samples or create your own
- Data Transfer: cut, copy, paste, drag & drop
- Internationalization: supports different input language, right to left reading
- Accessibility API: for people with disabilities
- Undo Framework API: supports unlimited numbers of actions to undo and redo
- Flexible Deployment: run within a browser as an applet or Java Web Start

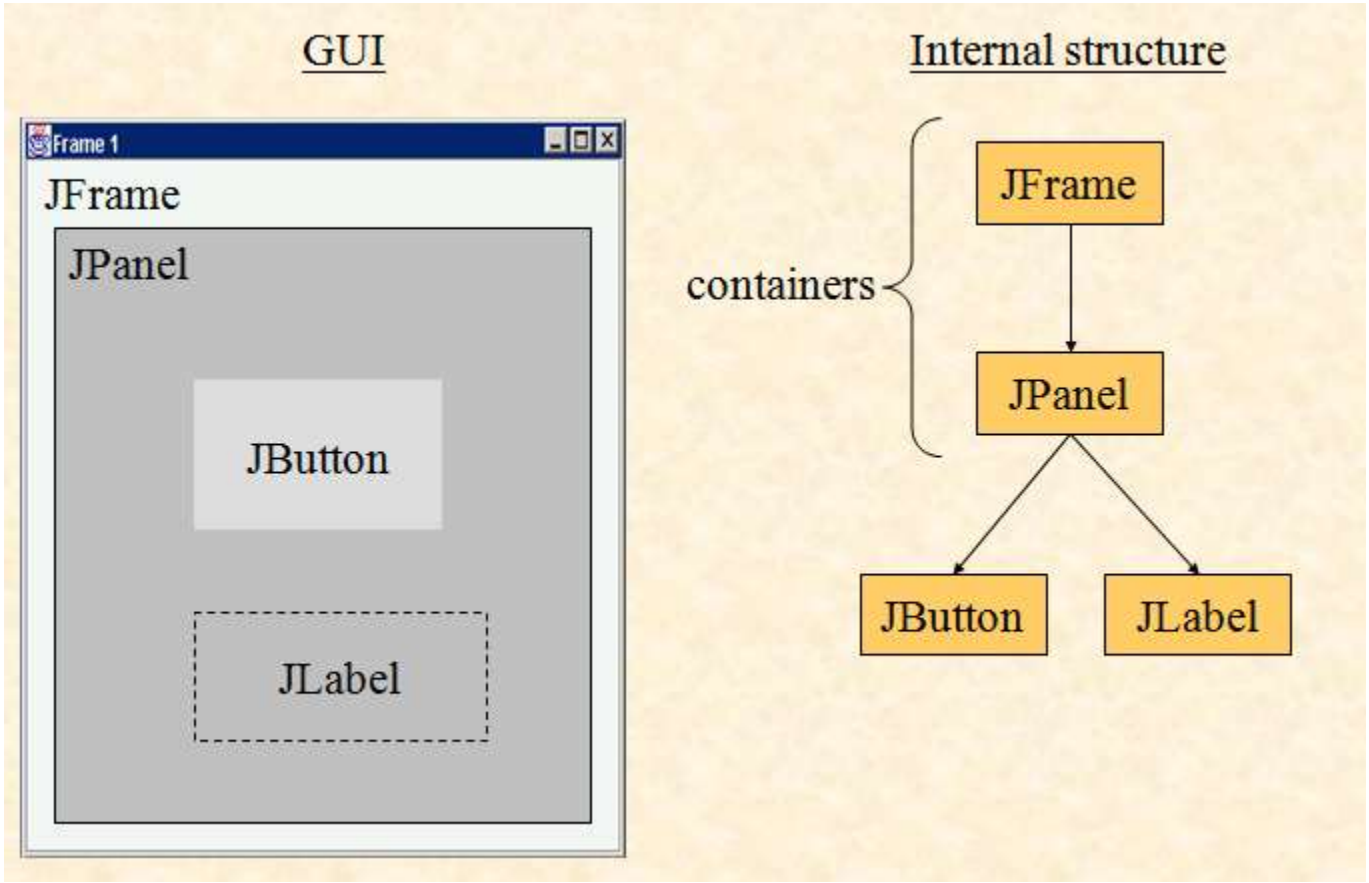
# Hierarchy of Swing Class



# Hierarchy of Swing Class

- Object: All classes ultimately derive from Object, thus this class is at the top of the tree.
- Component: represents an object that has a visual representation that can be shown on-screen and that can interact with users. This class defines some basic methods that are available to all Swing classes.
- Container: builds on the basic visual capabilities of the Component class by adding the ability to hold other containers.
- Window: a specialized type of container object that has a border, a title bar, buttons that minimize, maximize, and close the window, and that can be repositioned and possibly even resized by the user.

# Anatomy of Application GUI



# Components of Swing

- The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.
- **Swing follows MVC**(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.

## Commonly used Methods of Component class

The methods of Component class are widely used in java swing that are given below.

| Method | Description |
|---|---|
| public void add(Component c) | add a component on another component. |
| public void setSize(int width,int height) | sets size of the component. |
| public void setLayout(LayoutManager m) | sets the layout manager for the component. |
| public void setVisible(boolean b) | sets the visibility of the component. It is by default false. |

# Example of Swing

- There are two ways to create a frame:
- By creating the object of Frame class (association)
- By extending Frame class (inheritance)
- We can write the code of swing inside the main(), constructor or any other method.

## By creating the object of Frame class (association)

```
import javax.swing.*;
public class SimpleSwingExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Somple Frame");
        JButton jbutton = new JButton("CLICK");
        jbutton.setBounds(130,100,100, 40);
        frame.add(jbutton);
        frame.setSize(400, 300);
        frame.setLayout(null);
        frame.setVisible(true);
    }
}
```

# By extending Frame class (inheritance)

```
public class SimpleSwingExample2 extends JFrame{

    SimpleSwingExample2()
    {
      JButton b = new JButton("Click");
      b.setBounds(130,100,100, 40);
      add(b);
      setSize(400,500);
      setTitle("Simple Frame");
      setLayout(null);
      setVisible(true);

    }
    public static void main(String[] args) {
      JFrame frame = new SimpleSwingExample2();
   }
```

# Event and Listener (Java Event Handling)

- Changing the state of an object is known as an event.
- For example, click on button, dragging mouse etc.
- The java.awt.event package provides many event classes and Listener interfaces for event handling.
- Important Events and Action Listener Interfaces in Java:

| Event Classes | Listener Interfaces |
|---|---|
| ActionEvent | ActionListener |
| MouseEvent | MouseListener and MouseMotionListener |
| MouseWheelEvent | MouseWheelListener |
| KeyEvent | KeyListener |

# Java ActionListener Interface

- The Java ActionListener is notified whenever you click on the button or menu item.
- It is notified against ActionEvent.
- The ActionListener interface is found in java.awt.event .
- It has only one method: actionPerformed().
- The actionPerformed() method is invoked automatically whenever you click on the registered component.

**public abstract void** actionPerformed(ActionEvent e);

# The common approach is to implement the ActionListener.

1) Implement the ActionListener interface in the class:

**public class** ActionListenerExample Implements ActionListener

2) Register the component with the Listener:

component.addActionListener(instanceOfListenerclass);

3) Override the actionPerformed() method:

```
public void actionPerformed(ActionEvent e){
        //Write the code here
}
```

# Example of Event Handling in Java Swing by Anonymous class

```
public class JavaSwingExample3 extends JFrame
{

   public JavaSwingExample3() {

      JTextField jtf = new JTextField();
      jtf.setBounds(130, 50, 150, 40);
      JButton b = new JButton("Click Here");
      b.setBounds(130,100,100, 40);
      b.setBackground(Color.yellow);

      b.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent
e) {
            jtf.setText("Welcome to CSE2201.");
         }

      });

      add(jtf);
      add(b);

      setSize(400, 400);
      setLayout(null);
      setBackground(Color.RED);

   }
   public static void main(String[] args) {
      JavaSwingExample3 jse = new
     JavaSwingExample3();
      jse.setVisible(true);
   }


}
```

# Conclusion

- Java Swing is easier to learn than others because it’s Java
- You can use any helpful tools out there that are for Java development like eclipse IDE, NetBeans IDE
- Lacks live graphical and interactive help while developing
- Has unlimited possibilities depending on how you implement your software