Graphical User Interfaces

- AWT and Swing packages
- Frames and Panels
- Components
- Nested Panels
- Images
- Reading for this Lecture: L&L, 3.9 – 3.11
Graphical Applications

• The example programs we've explored thus far have been text-based *command-line applications*, which interact with the user using text prompts.

• Let's examine some Java applications that have graphical components based on objects.

• These components will serve as a foundation for programs with graphical user interfaces (GUIs).

• The important point to learn is that an object in your program corresponds with a real world object.

• We will start by generating a frame with panels containing text “labels” or images.
Graphical Applications

- GUI-related classes are defined primarily in `java.awt` and `javax.swing` packages.
- The Abstract Windowing Toolkit (AWT) was the original Java GUI package.
- The Swing package provides additional and more versatile components.
- Sometimes called Java Foundation Classes (mimicking Microsoft Foundation Classes).
GUI Containers - Frame

• A GUI container is a component that is used to hold and organize other components
• JFrame, JDialog, and JApplet are the three top level containers that are used to display graphics in GUI applications
• We will work only with JFrame for now
• A JFrame is displayed as a separate window with a title bar – it can be repositioned and resized on the screen as needed
Frame-based Hello World

JFrame frame

JFrame frame attribute title

JFrame frame width = 300

JFrame frame height = 200

JLabel label

Hello World
import javax.swing.*;    // Get JFrame and JLabel classes

public class HelloWorld {
    public static void main(String[] args) {
        // Create and set up the window with a frame object
        JFrame frame = new JFrame("HelloWorld Using Swing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Add the ubiquitous "Hello World" label to the frame object
        JLabel label = new JLabel("Hello World");
        frame.add(label);

        // Display the window using frame methods
        frame.setSize(300,200);  // width and height
        frame.setVisible(true);
    }
}

Frame-based Hello World
GUI Panels / Components

• A *panel* is a container that cannot be displayed on its own
• It must be added to a top level container or another panel to be displayed
• It is used to organize other components
• A *GUI component* is an object that represents a screen element such as a text field or an image
• A GUI component must be added to another container such as a frame or panel to be displayed
Labels

• A *label* is a GUI component that displays a line of text
• Labels are usually used to display information or identify other components in the display
• Let's look at a program that organizes two labels in a panel and displays that panel in a frame
• See *Authority.java* (page 145)
• This program is not interactive, but the frame can be repositioned and resized
Authority

Question authority, 
but raise your hand first.
import java.awt.*;
import javax.swing.);

public class Authority
{
    public static void main (String[] args)
    {
        JFrame frame = new JFrame ("Authority");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel primary = new JPanel();
        primary.setBackground (Color.YELLOW);
        primary.setPreferredSize (new Dimension(250, 75));
Authority.java

```java
JLabel label1 = new JLabel ("Question authority,");
JLabel label2 = new JLabel ("but raise your hand first.");

primary.add (label1); // add the labels to the panel
primary.add (label2);

frame.add(primary); // add the panel to the frame
frame.pack(); // set frame size based on contents
frame.setVisible(true);
```
```
Nested Panels

• Containers that contain other components make up the containment hierarchy of an interface
• This hierarchy can be as intricate as needed to create the visual effect desired
• The following example nests two panels inside a third panel – note the effect this has as the frame is resized
• See NestedPanels.java (page 148-149)
Nested Panels

JPANEL panel1

JLabel label1

JLabel label2

JLabel label3

One

Two

Buckle my shoe ...
import java.awt.*;
import javax.swing.*;

public class NestedPanels
{
    public static void main (String[] args)
    {
        JFrame frame = new JFrame ("Nested Panels");
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        // Set up first subpanel
        JPanel subPanel1 = new JPanel();
        subPanel1.setPreferredSize (new Dimension(150, 100));
        subPanel1.setBackground (Color.GREEN);
        JLabel label1 = new JLabel ("One");
        subPanel1.add (label1);
NestedPanels.java

// Set up second subpanel
JPanel subPanel2 = new JPanel();
subPanel2.setPreferredSize (new Dimension(150, 100));
subPanel2.setBackground (Color.RED);
JLabel label2 = new JLabel("Two");
subPanel2.add (label2);

// Set up primary panel
JPanel primary = new JPanel();
primary.setBackground (Color.YELLOW);
primary.add (subPanel1);
primary.add (subPanel2);
JLabel label3 = new JLabel("Buckle my shoe ...");
primary.add (label3);

frame.add(primary);
frame.pack();
frame.setVisible(true);
Images

- Images are often used in programs with a graphical interface.
- Java can manage images in both JPEG and GIF formats.
- As we've seen, a JLabel object can be used to display a line of text.
- It can also be used to display an image.
- That is, a label can be composed of text, and image, or both at the same time.
Images

• The `ImageIcon` class is used to represent an image that is stored in a label.
• The position of the text relative to the image can be set explicitly.
• The alignment of the text and image within the label can be set as well.
• See `LabelDemo.java` (page 151)
LabelDemo

Devil Left

Devil Right

Devil Above
import java.awt.*;
import javax.swing.*;

public class LabelDemo
{
    public static void main (String[] args)
    {
        JFrame frame = new JFrame ("Label Demo");
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        ImageIcon icon = new ImageIcon ("devil.gif");

        JLabel label1, label2, label3;

        label1 = new JLabel ("Devil Left", icon, SwingConstants.CENTER);

        label2 = new JLabel ("Devil Right", icon, SwingConstants.CENTER);
        label2.setHorizontalTextPosition (SwingConstants.LEFT);
        label2.setVerticalTextPosition (SwingConstants.BOTTOM);
label3 = new JLabel("Devil Above", icon, SwingConstants.CENTER);
label3.setHorizontalTextPosition(SwingConstants.CENTER);
label3.setVerticalTextPosition(SwingConstants.BOTTOM);

JPanel panel = new JPanel();
panel.setBackground(Color.CYAN);
panel.setPreferredSize(new Dimension(200, 250));
panel.add(label1);
panel.add(label2);
panel.add(label3);

frame.add(panel);
frame.pack();
frame.setVisible(true);