CSE 11

February 27, 2003
Quiz

Give the headings for all the methods required by the ActionListener Interface.
ICONS

Icon Demonstration

Nice day!

Glad to meet you!

Hello  Goodbye
ICONS

In Labels
JLabel niceLabel = new JLabel("Nice day!");
ImageIcon smileyIcon =
    new ImageIcon("smiley.gif");
niceLabel.setIcon(smileyIcon);

In Buttons
JButton helloButton = new JButton("Hello");
ImageIcon dukeWavingIcon =
    new ImageIcon("duke_waving.gif");
helloButton.setIcon(dukeWavingIcon);
Scroll Bars

![Scrolling Memo Saver](image)
Scroll Bars

```java
JTextArea theText =
    new JTextArea(LINES, CHAR_PER_LINE);
JScrollPane scrolledText =
    new JScrollPane(theText);
textPanel.add(scrolledText);
```

Note that, in some sense, the text area is added to the scroll bars; not, the other way round.
Borders
Read on Your Own
BoxLayout Manager and the Box class

Mostly Read on your Own
BoxLayout Manager
Horizontal and Vertical versions
Horizontal Version is like FlowLayout
Manager but allows:
Struts and Glue.

Strut == ridged separator
Glue == flexible separator
Box Class
Is Like a JPanel with a BoxLayout Manager
CardLayout Manager
Allows you to have multiple views in a container and go from one view to another, like going from one card in a deck to another.

Read on your own.
public interface WindowListener
{
    public void windowOpened(WindowEvent e);
    public void windowClosing(WindowEvent e);
    public void windowClosed(WindowEvent e);
    public void windowIconified(WindowEvent e);
    public void windowDeiconified(WindowEvent e);
    public void windowActivated(WindowEvent e);
    public void windowDeactivated(WindowEvent e);
    public void windowGainedFocus(WindowEvent e);
    public void windowLostFocus(WindowEvent e);
    public void windowStateChanged(WindowEvent e);
}
public class WindowListenerDemo extends JFrame
        implements ActionListener,
                WindowListener
{
    public WindowListenerDemo()
    {
        setSize(WIDTH, HEIGHT);
        ....
        addWindowListener(this);
        .....  
    }

    public void actionPerformed(ActionEvent e)
    {
        ....
    }

    public void windowOpened(WindowEvent e)
    {
    }

    public void windowClosing(WindowEvent e)
    {
        this.dispose();
        System.exit(0);
    }
}
public void windowClosed(WindowEvent e) {}

public void windowIconified(WindowEvent e) {}

public void windowDeiconified(WindowEvent e) {}

public void windowActivated(WindowEvent e) {}

public void windowDeactivated(WindowEvent e) {}

public void windowGainedFocus(WindowEvent e) {}

public void windowLostFocus(WindowEvent e) {}

public void windowStateChanged(WindowEvent e) {}
}
Multi-Window GUIs

Are you sure you want to exit?

Yes  No

Please don't click that button.
public class CloseWindowDemo extends JFrame {
    ....
    public CloseWindowDemo( ) {
        setSize(WIDTH, HEIGHT);
        setDefaultCloseOperation(
            WindowConstants.DO_NOTHING_ON_CLOSE);
        addWindowListener(new InnerDestroyer( ));
        setTitle("Close Window Demo");
        Container contentPane = getContentPane( );
        contentPane.setLayout(new BorderLayout( ));
        JLabel message =
            new JLabel("Please don't click button.");
        contentPane.add(message, BorderLayout.CENTER);
    }

    private class InnerDestroyer extends WindowAdapter {
        public void windowClosing(WindowEvent e) {
            ConfirmWindow askWindow =
                new ConfirmWindow( );
            askWindow.setVisible(true);
        }
    }
}
private class ConfirmWindow extends JFrame implements ActionListener {
    ...
    public ConfirmWindow( ) {
        ..... 
        JPanel buttonPanel = new JPanel( );
        buttonPanel.setLayout(new FlowLayout( ));

        JButton exitButton = new JButton("Yes");
        exitButton.addActionListener(this);
        buttonPanel.add(exitButton);

        JButton cancelButton = new JButton("No");
        cancelButton.addActionListener(this);
        buttonPanel.add(cancelButton);

        confirmContent.add(
            buttonPanel, BorderLayout.SOUTH);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand( ).equals("Yes"))
            System.exit(0);
        else if (e.getActionCommand( ).equals("No"))
            dispose( ); // Destroys ConfirmWindow.
    }
}
All of Chapter 14 is covered on Final, even if not all covered in lecture.