



Graphical User Interfaces (GUIs)

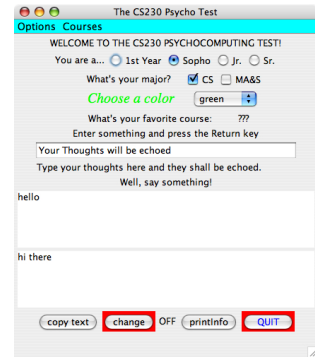
Reading: <http://docs.oracle.com/javase/tutorial/uiswing/>

L - 1



Da GUI

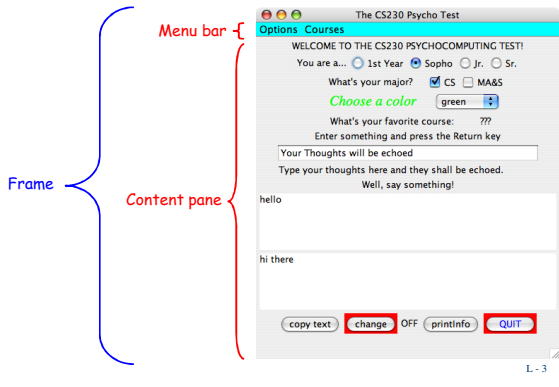
- Users demand nice looks (and maybe efficient programs)
- Composed of a few simple Jcomponents:
 - JMenuBar
 - JLabel
 - JButton
 - JRadioButton
 - JCheckBox
 - JTextField
 - JComboBox
 - JTextArea



L - 2



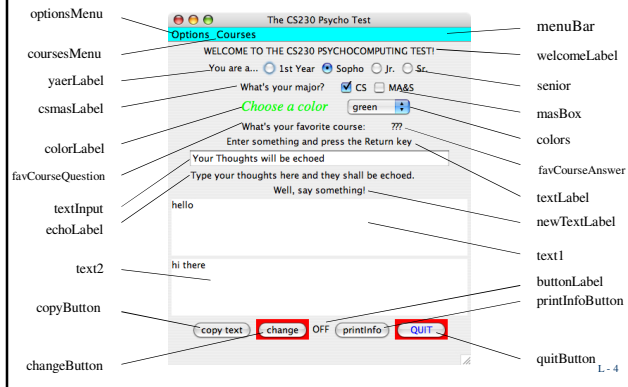
The Frame (or Applet) contains a menu bar and content pane



L - 3



Component Variables



L - 4

Instance Variables for Components

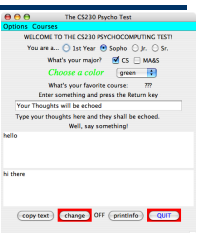
```

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

public class GUIDemo extends JApplet
    implements ActionListener, ItemListener {

    private JMenuBar menuBar;
    private JMenuItem quitMenuItem;
    private JMenuItem CS111, CS215, CS230, CS231, CS235, CS240;

    private JLabel welcomeLabel, textLabel, newTextLabel;
    private JLabel echoLabel, csmasLabel;
    private JLabel yearLabel, colorLabel, buttonLabel;
    private JLabel favCourseQuestion, favCourseAnswer;
    private JButton copyButton, changeButton;
    private JButton printInfoButton, quitButton;
    private JRadioButton firstYear, sopho, junior, senior;
    private JCheckBox csBox, masBox;
    private ButtonGroup numbers;
    private JTextField textInput;
    private JComboBox colors;
    private JTextArea text1, text2;
    
```



```

// constructor
public GUIDemo() {
    makeMenuBar();
    makeLabels();
    makeButtons();
    makeRadiobuttons();
    makeCheckboxes();
    makeTextField();
    makeComboBox();
    makeTextAreas();
}

```

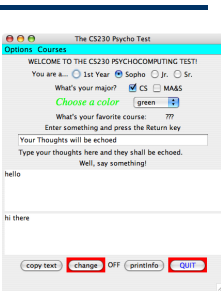
L-5

public void makeLabels() {

```

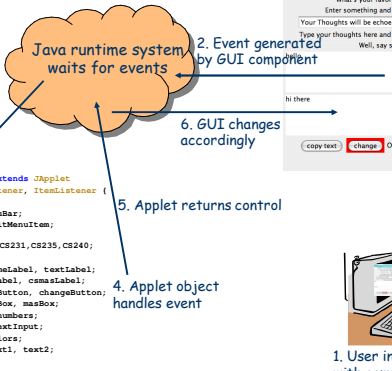
// creates all of the Labels used in the GUI
welcomeLabel = new JLabel("WELCOME TO THE CS230
    PSYCHOCOMPUTING TEST!");
csmasLabel = new JLabel("What's your major?");
textLabel = new JLabel("Enter something and press the
    Return key");
echoLabel = new JLabel("Type your thoughts here and they
    shall be echoed.");
newTextLabel = new JLabel("Well, say something!");
buttonLabel = new JLabel("OFF");
yearLabel = new JLabel("You are a . . .");
favCourseQuestion = new JLabel("What's your favorite
    course:");
favCourseAnswer = new JLabel(" ??? ");
colorLabel = new JLabel("Choose a color");
// every GUI component class inherits the
// setForeground() method from the JComponent superclass
colorLabel.setForeground(Color.green);
colorLabel.setFont(new Font("Times", Font.ITALIC, 20));
}

```



L-6

Handling Events



```

public class GUIDemo extends JApplet
    implements ActionListener, ItemListener {

    private JMenuBar menuBar;
    private JMenuItem quitMenuItem;
    private JMenuItem
        CS111, CS215, CS230, CS231, CS235, CS240;

    private JLabel welcomeLabel, textLabel;
    private JLabel echoLabel, csmasLabel;
    private JButton copyButton, changeButton;
    private JCheckBox csBox, masBox;
    private ButtonGroup numbers;
    private JTextField textInput;
    private JComboBox colors;
    private JTextArea text1, text2;
    ...
}

```

L-7

public void makeButtons() {

```

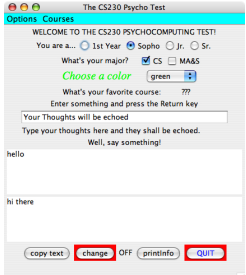
// creates the JButtons used in the GUI
changeButton = new JButton("change");
// we add an Action Listener to this button
// so we can catch and react to clicks on it
changeButton.addActionListener(this);
changeButton.setBackground(Color.red);

printInfoButton = new JButton("printInfo");
printInfoButton.addActionListener(this);

quitButton = new JButton("QUIT");
quitButton.addActionListener(this);
quitButton.setBackground(Color.red);
quitButton.setForeground(Color.blue);

copyButton = new JButton("copy text");
copyButton.addActionListener(this);
}

```



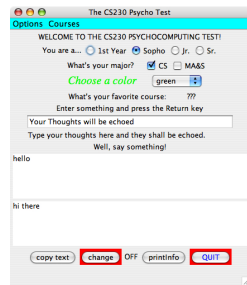
L-8



public void makeRadioButtons () {

```
// these four JRadioButtons are grouped
// in a ButtonGroup
numbers = new ButtonGroup();
firstYear = new JRadioButton("1st Year");
firstYear.addActionListener(this);
sopho = new JRadioButton("Sopho");
sopho.addActionListener(this);
sopho.setSelected(true);
junior = new JRadioButton("Jr.");
junior.addActionListener(this);
senior = new JRadioButton("Sr.");
senior.addActionListener(this);
numbers.add(firstYear);
numbers.add(sopho);
numbers.add(junior);
numbers.add(senior);
```

}



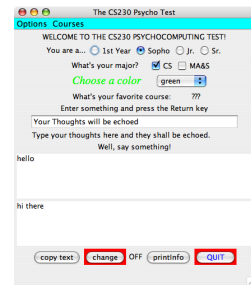
L - 9



public void makeComboBox () {

```
colors = new JComboBox ();
colors.addItem("red");
colors.addItem("orange");
colors.addItem("yellow");
colors.addItem("blue");
colors.addItem("green");
colors.addItem("black");
colors.setSelectedIndex(4);
colors.addActionListener(this);
```

}



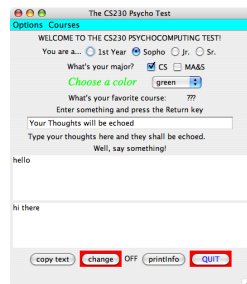
L - 10



public void makeTextField () {

```
textInput = new JTextField("Your Thoughts will
be echoed", 30);
textInput.addActionListener(this);
```

}



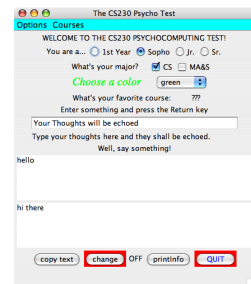
L - 11



public void makeTextArea () {

```
text1 = new JTextArea("hello", 5, 35);
text2 = new JTextArea("hi there", 5, 35);
```

}



L - 12

DATA TRU `public void makeCheckBoxes() {`

```

csBox = new JCheckBox("CS", true);
csBox.addItemListener(this);

masBox = new JCheckBox("MA&S", false);
masBox.addItemListener(this);
}

```

L - 13

DATA TRU `public void makeMenuBar() {`

```

menuBar = new JMenuBar();
menuBar.setOpaque(true);
menuBar.setBackground(Color.cyan);
menuBar.setPreferredSize(new Dimension(200, 20));

// Make first of two menus in menu bar
JMenu optionsMenu = new JMenu("Options");
menuBar.add(optionsMenu); // Add first menu to menu bar
optionsMenu.setBackground(Color.cyan);
quitMenuItem = new JMenuItem("QUIT");
optionsMenu.add(quitMenuItem); // Add first menu item to menu
quitMenuItem.addActionListener(this); // Listen to see if menu
// item is selected

// Make second of two menus in menu bar
JMenu coursesMenu = new JMenu("Courses");
menuBar.add(coursesMenu); // Add second menu to menu bar
coursesMenu.setBackground(Color.cyan);
CS111 = new JMenuItem("CS111");
coursesMenu.add(CS111); // Add first menu item to menu
CS111.addActionListener(this); // Listen to see if menu
// item is selected
coursesMenu.addSeparator();
...
}

```

L - 14

DATA TRU `public void init() {`

```

// Sets menu bar.
// Adds each GUI component to the applet.
// The appearance of the GUI depends on the order that components are added
// and on the size of the frame that will contain them

this.setJMenuBar(menuBar);

this.setLayout(new FlowLayout()); // Start with simple FlowLayout

add(welcomeLabel);          add(favCourseQuestion);
add(favCourseAnswer);

add(yearLabel);            add(textLabel);
add(firstYear);           add(favCourseAnswer);
add(sopho);               add(textInput);
add(junior);              add(echoLabel);
add(senior);              add(newTextLabel);

add(camasLabel);         add(text1);
add(caBox);              add(text2);
add(masBox);

add(colorLabel);        add(copyButton);
add(colors);            add(changeButton);
                        add(buttonLabel);
                        add(printInfoButton);
                        add(quitButton);
}

```

L - 15

DATA TRU Handling Action Events

```

public void actionPerformed(ActionEvent event)
{
// This method implements the ActionListener
// interface and must be defined to handle
// the events of a Button click or pressing
// the "Enter" or "Return" key in a TextField

Object source = event.getSource();

if (source == quitMenuItem) {
// user selected QUIT from menu
System.exit(0);
} else if (source.equals(firstYear)) {
// user pushed radio button
yearLabel.setText("Fresh...");
} else if (source.equals(sopho)) {
// user pushed radio button
yearLabel.setText("Wise!");
} else if (source.equals(junior)) {
// user pushed radio button
yearLabel.setText("Wiser!");
} else if (source.equals(senior)) {
// user pushed radio button
yearLabel.setText("Senioritis!");
} else if ...
} else if (source == quitButton) {
// user pushed quitButton
System.exit(0);
} else if (source.equals(changeButton)) {
// user pushed the changeButton
if ((buttonLabel.getText()).equals("ON")) {
buttonLabel.setText("OFF");
changeButton.setBackground(Color.blue);
printInfoButton.setEnabled(false);
} else {
buttonLabel.setText("ON");
changeButton.setBackground(Color.red);
printInfoButton.setEnabled(true);
}
} else if (source.equals(printInfoButton)) {
// user pushed the printInfoButton
printGuiInfo();
text2.setText("Printed GUI info \n in
console window");
} else if ...
}

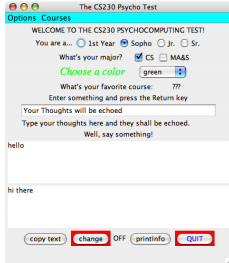
```

L - 16



Item Selection Events

```
public void itemStateChanged (ItemEvent event) {
    // This method implements the ItemListener interface and must be
    // defined for handling a change made to a JCheckbox
    Object source = event.getSource();
    if (source.equals(csBox))
        csmasLabel.setText("cool!");
    else if (source.equals(masBox))
        csmasLabel.setText("wow!");
}
```



L - 17



Your Code is Executed as Usual

```
public void printGuiInfo () {
    // prints info about the status of JCheckboxes,
    // JRadioButtons,
    // and JList
    System.out.println("csBox: " + csBox.isSelected());
    System.out.println("masBox: " + masBox.isSelected());
    System.out.println("firstYear: " + firstYear.isSelected());
    System.out.println("sopho: " + sopho.isSelected());
    System.out.println("junior: " + junior.isSelected());
    System.out.println("senior: " + senior.isSelected());
    int index = colors.getSelectedIndex();
    String color = colors.getSelectedItem().toString();
    System.out.println(color + " selected at index " + index);
}
```

L - 18



Create the GUI and show it

```
private static void createAndShowGUI() {
    JFrame.setDefaultLookAndFeelDecorated(true);
    JFrame frame = new JFrame("The CS230 Psycho Test");
    frame.setSize(430, 500);

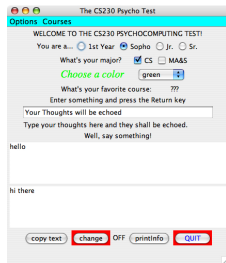
    // Specify that the close button exits application
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    GUIDemo applet = new GUIDemo();
    applet.init();

    frame.add(applet, BorderLayout.CENTER);

    frame.setVisible(true); // Display the window
}

public static void main(String[] args) {
    // Schedule a job for the event-dispatching thread:
    // creating and showing this application's GUI
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() { createAndShowGUI(); }
    });
}
```

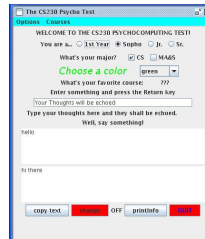


Note: these 2 class methods are only needed to run the applet as an application L - 19



Application vs. Applet

```
[btjaden@puma] java GUIDemo & <applet code=GUIDemo.class
width=385 height=500>
</applet>
```



L - 20

ERROR: stackunderflow
OFFENDING COMMAND: ~

STACK: