

Graphical User Interface (GUI)

CPSC 1181 – O.O.

Jeremy Hilliker
Summer 2017

Langara.

THE COLLEGE OF HIGHER LEARNING.

Overview

- Concepts
 - Component
 - Container
 - Layout Manager
- Controls
- Layout planning
- Menus

Concepts

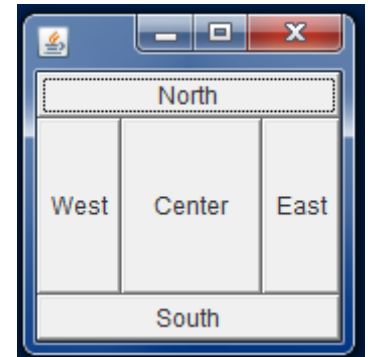
- UI Component:
 - A “thing” in the GUI
 - Button, Label, TextField, Panel, ScrollPane
- UI Container:
 - A thing that holds components
- Layout Manager
 - Does the layout of components within a container
 - Flow, Border, Grid, Box, etc
- UIs are composed of many Components, Containers, and Layouts

Layout

- By default, JPanel uses FlowLayout
 - Arranges left to right, starts new row when full
- You can set others:
`panel.setLayout(new BorderLayout())`

Border Layout

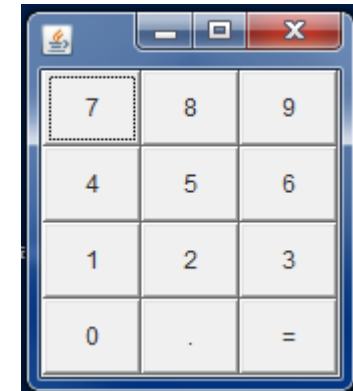
```
4 // javac *.java; java -ea BorderLayoutEx
5 public class BorderLayoutEx extends JFrame {
6     public static void main(String[] args) {
7         JFrame frame = new JFrame();
8         frame.setSize(500, 400);
9         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10
11         Container pane = frame.getContentPane();
12         assert pane.getLayout() instanceof BorderLayout;
13
14         pane.add(new Button("North"), BorderLayout.NORTH);
15         pane.add(new Button("South"), BorderLayout.SOUTH);
16         pane.add(new Button("East"), BorderLayout.EAST);
17         pane.add(new Button("West"), BorderLayout.WEST);
18         pane.add(new Button("Center"), BorderLayout.CENTER);
19
20         frame.setVisible(true);
21     }
22 }
```



GridLayout

- Arranges components in a grid of fixed rows and columns
- All components become same size
- Add: left-to-right, top-to-bottom

GridLayout

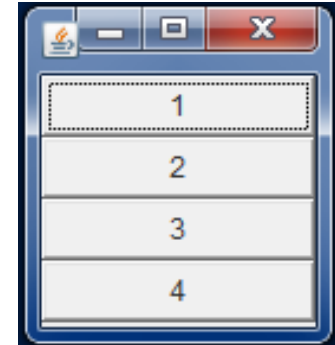


```
4 // javac GridLayoutEx.java; java -ea GridLayoutEx
5 public class GridLayoutEx extends JFrame {
6     public static void main(String[] args) {
7         JFrame frame = new JFrame();
8         frame.setSize(500, 400);
9         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10
11         Container pane = frame.getContentPane();
12         pane.setLayout(new GridLayout(4, 3));
13
14         final char[] labels = new char[]
15             {'7', '8', '9', '4', '5', '6', '1', '2', '3', '0', '.', '='};
16
17         for(char l : labels) {
18             pane.add(new Button(Character.toString(l)));
19         }
20
21         frame.setVisible(true);
22     }
23 }
```

BoxLayout

- Aligns components vertically (stacks) or horizontally
- See also: glue

BoxLayout



```
4 // javac BoxLayoutEx.java; java -ea BoxLayoutEx
5 public class BoxLayoutEx extends JFrame {
6     public static void main(String[] args) {
7         JFrame frame = new JFrame();
8         frame.setSize(500, 400);
9         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10
11         Container pane = frame.getContentPane();
12         pane.setLayout(new BoxLayout(pane, BoxLayout.Y_AXIS));
13
14         final char[] labels = new char[]
15             {'1', '2', '3', '4'};
16
17         for(char l : labels) {
18             pane.add(new Button(Character.toString(l)));
19         }
20
21         frame.setVisible(true);
22     }
23 }
```

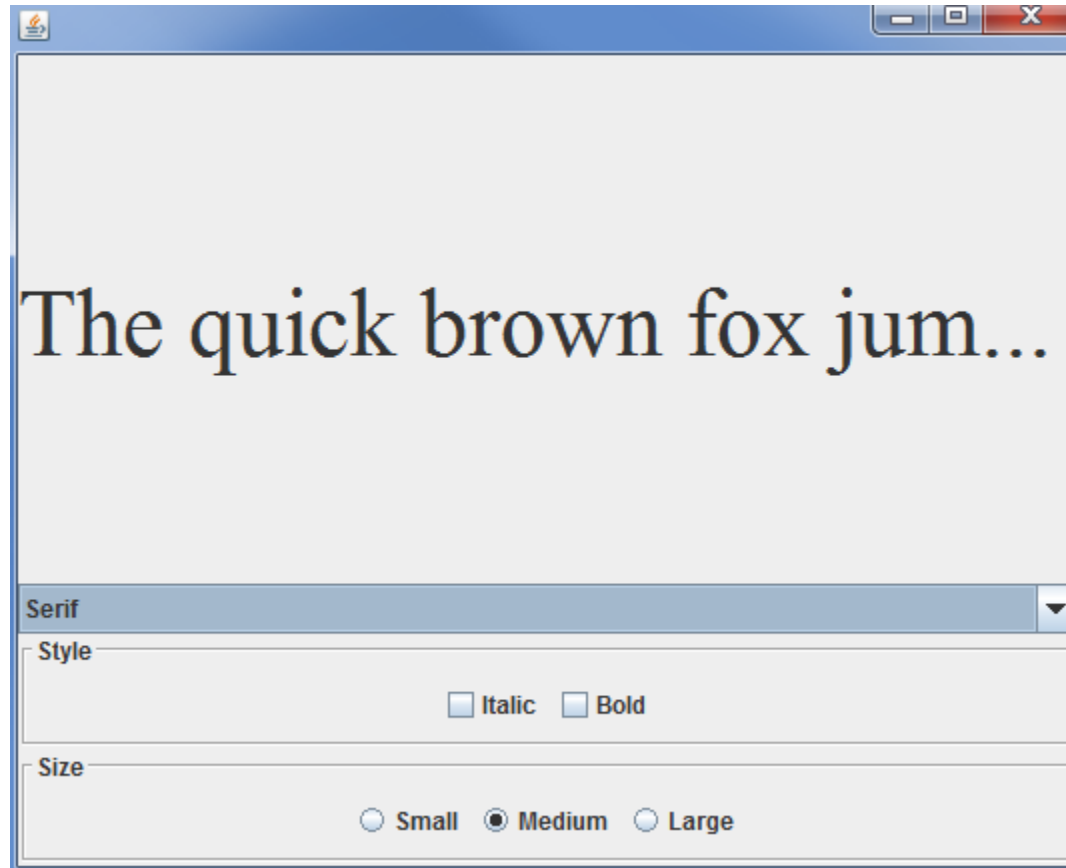
GridBagLayout

- Tabular arrangement of components
 - Like a table
 - Columns can have different sizes
 - Components can span multiple rows
- Complicated
 - For interest:
<https://docs.oracle.com/javase/tutorial/uiswing/layout/gridbag.html>

UIs

- Create a UI by making compositions of many panels, layouts, and components
- Respond to user actions through events and event listeners

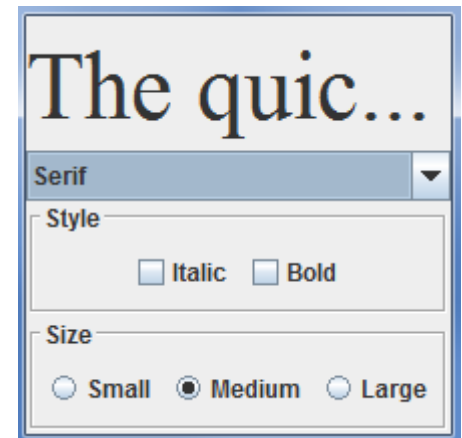
UI Controls



```

6 // javac UIExFrame.java; java -ea UIExFrame
7 public class UIExFrame extends JFrame {
8
9     private final static String sampleText =
10         "The quick brown fox jumped over the lazy dog.";
11
12     private final JLabel label;
13     private final JComboBox<String> combo;
14     private final JCheckBox italics;
15     private final JCheckBox bold;
16     private final ButtonGroup size;
17     private final JRadioButton sml;
18     private final JRadioButton med;
19     private final JRadioButton lrg;
20
21     public UIExFrame() {
22         setLayout(new BorderLayout());
23
24         label = new JLabel(sampleText);
25         add(label, BorderLayout.CENTER);
26
27         ActionListener listener = new ActionListener() {
28             public void actionPerformed(ActionEvent e) {
29                 setLabel();
30             }
31         };

```



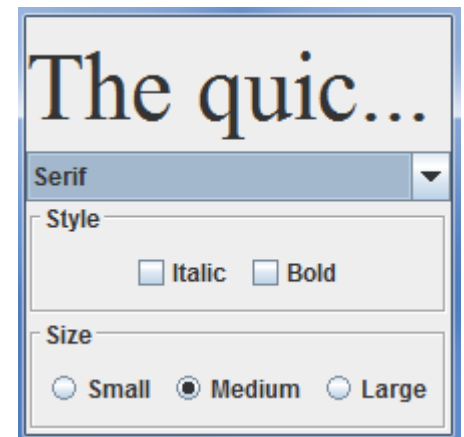
Events

- UI components fire events when their state changes
 - Button clicked, box selected, etc.
- Register a listener with the component to be notified

```

33 JPanel south = new JPanel();
34 south.setLayout(new BorderLayout(south, BorderLayout.Y_AXIS));
35
36 combo = new JComboBox<String>();
37 combo.addItem("Serif");
38 combo.addItem("SanSerif");
39 combo.addItem("Monospaced");
40 combo.addActionListener(listener);
41 south.add(combo);
42
43 JPanel p = new JPanel();
44 italics = new JCheckBox("Italic");
45 bold = new JCheckBox("Bold");
46 italics.addActionListener(listener);
47 bold.addActionListener(listener);
48 p.add(italics);
49 p.add(bold);
50 p.setBorder(new TitledBorder(new EtchedBorder(), "Style"));
51 south.add(p);

```



Combo Box

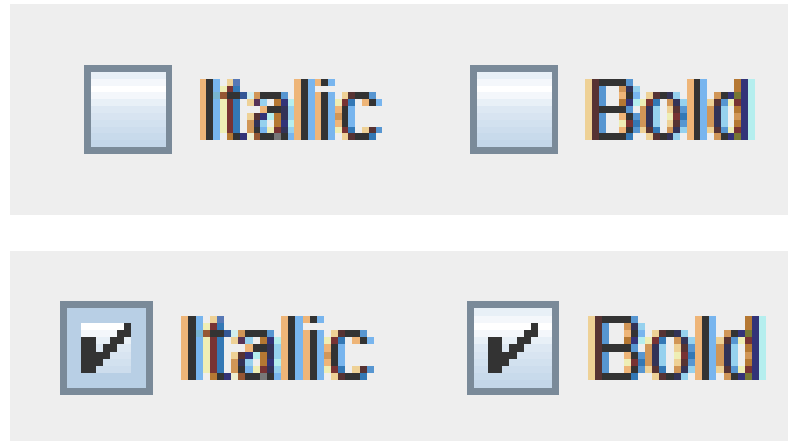
- For a large set of mutually exclusive options
- “Combo” because combination of
 - Text field
 - List
- Can allow user to type in field: `setEditable`



- `getSelectedItem`

Checkbox

- Two states: checked, unchecked
- Use for non-mutually-exclusive options

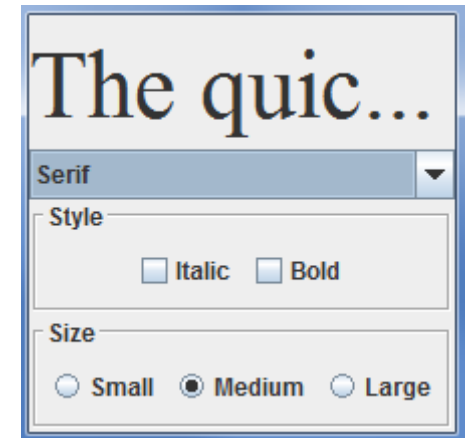


- `isSelected` / `setSelected`

```

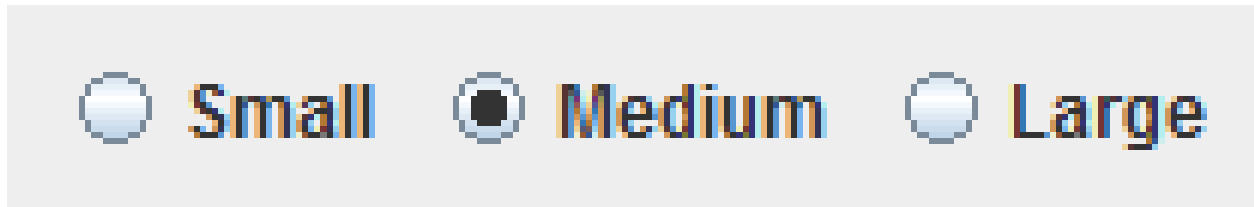
53 p = new JPanel();
54 size = new ButtonGroup();
55 sml = new JRadioButton("Small");
56 med = new JRadioButton("Medium");
57 lrg = new JRadioButton("Large");
58 sml.addActionListener(listener);
59 med.addActionListener(listener);
60 lrg.addActionListener(listener);
61 size.add(sml);
62 size.add(med);
63 size.add(lrg);
64 med.setSelected(true);
65 p.add(sml);
66 p.add(med);
67 p.add(lrg);
68 p.setBorder(new TitledBorder(new EtchedBorder(), "Size"));
69 south.add(p);
70
71 add(south, BorderLayout.SOUTH);
72
73 setLabel();
74 }

```



Radio Button

- For a small set of mutually exclusive options
- In a ButtonGroup, only one button can be selected at a time

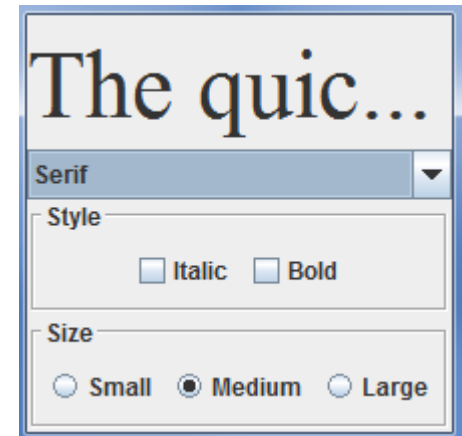


- `getSelectedItem` / `setSelectedItem`

```

76 private void setLabel() {
77     String face = (String) combo.getSelectedItem();
78
79     int style = 0;
80     if(italics.isSelected()) style += Font.ITALIC;
81     if(bold.isSelected()) style += Font.BOLD;
82
83     int size;
84     final int SML = 24;
85     final int MED = 48;
86     final int LRG = 72;
87     if(sml.isSelected()) size = SML;
88     else if(med.isSelected()) size = MED;
89     else size = LRG;
90
91     label.setFont(new Font(face, style, size));
92 }
93
94 public static void main(String[] args) {
95     JFrame frame = new UIExFrame();
96     frame.setSize(500, 400);
97     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
98     frame.setVisible(true);
99 }

```



Text Field

- A text input box in which to type
 - actionPerformed() on focus change or enter



The quick brown fox jumped over the lazy dog.

Text Area

- Input for multiple lines of text
 - Dimensions set on construction
 - Can setText / getText / append

Others

- Dialogs
- Choosers
- Labels
- Lists
- Scroll Panes
- Spinners
- Sliders
- Tabbed Panes
- Tables
- Toolbars
- Trees
- More...

Layout Management

Step 1: Make a sketch of your desired component layout

Size

☒ Small

☐ Medium

☐ Large

☒ Pepperoni

☒ Anchovies

Your Price:

Layout Management

Step 2: Find groupings of adjacent components with the same layout

Size

☒ Small

☐ Medium

☐ Large

☒ Pepperoni

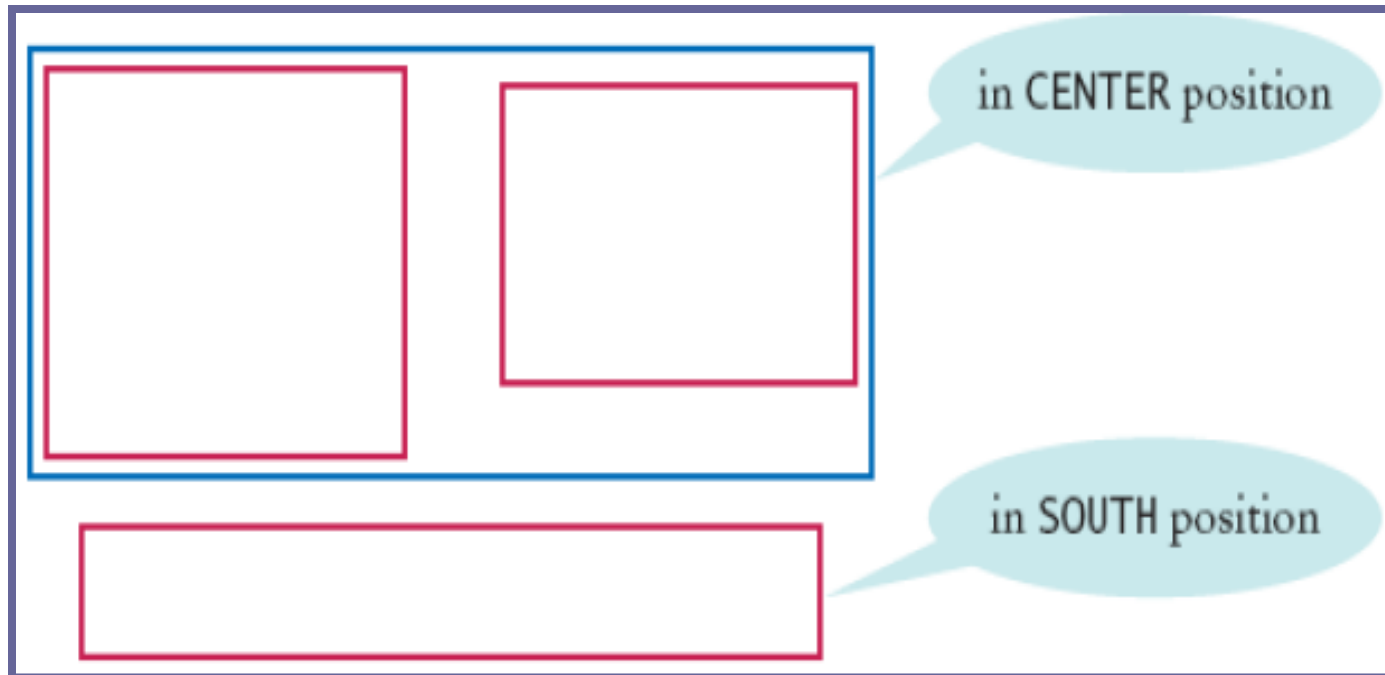
☒ Anchovies

Your Price:

Layout Management

Step 3: Identify layouts for each group

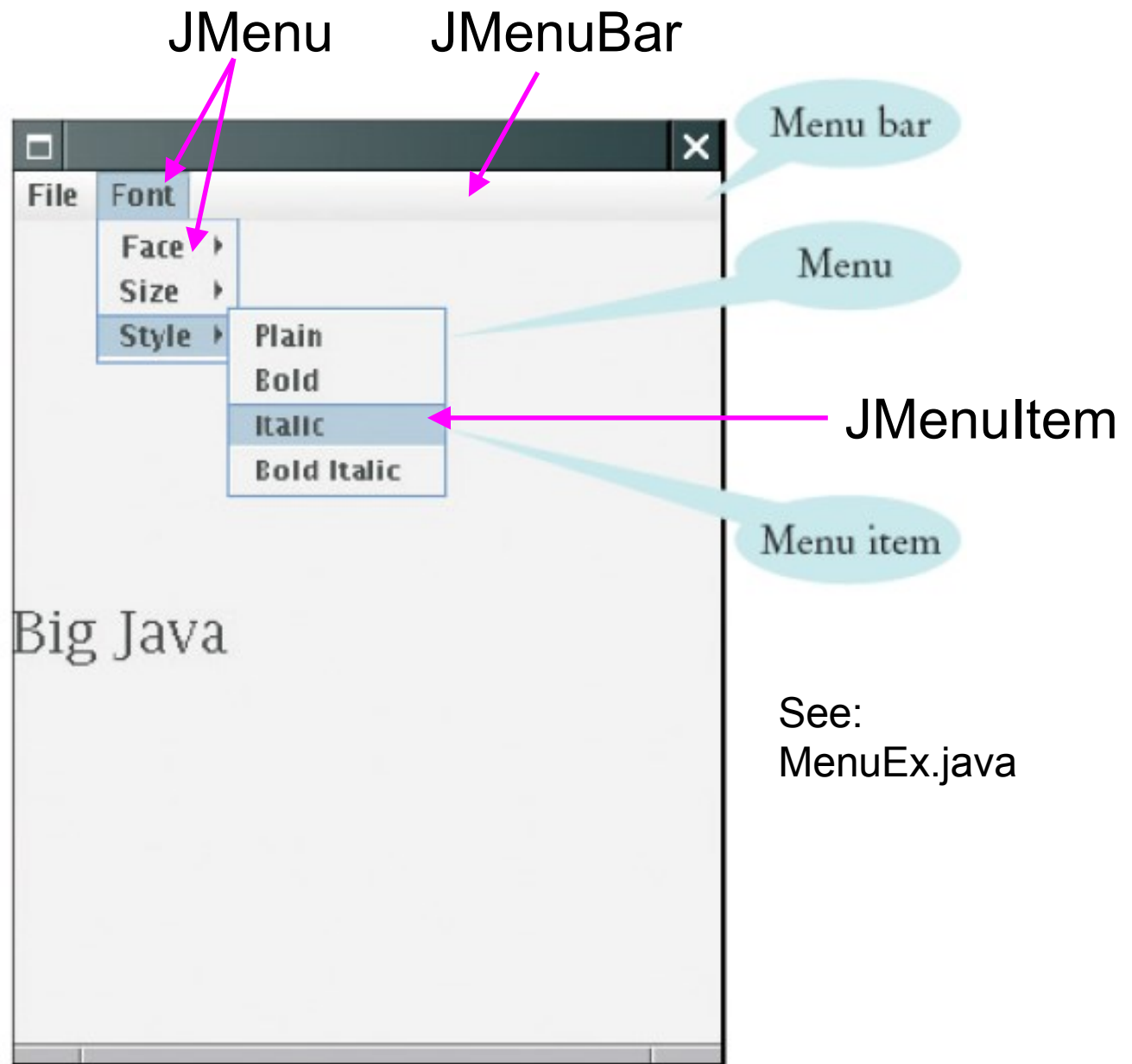
Step 4: Group the groups together



Step 5: Write the code to generate the layout

Menus

- A frame can have a menu bar (JMenuBar)
- A menu bar contains menus (JMenu)
- A menu contains submenus and menu items (JMenuItem)
 - Add your listeners only to the menu items, not the bar or menus



Recap

- Concepts
 - Component
 - Container
 - Layout Manager
- Controls
- Layout planning
- Menus