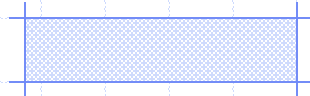
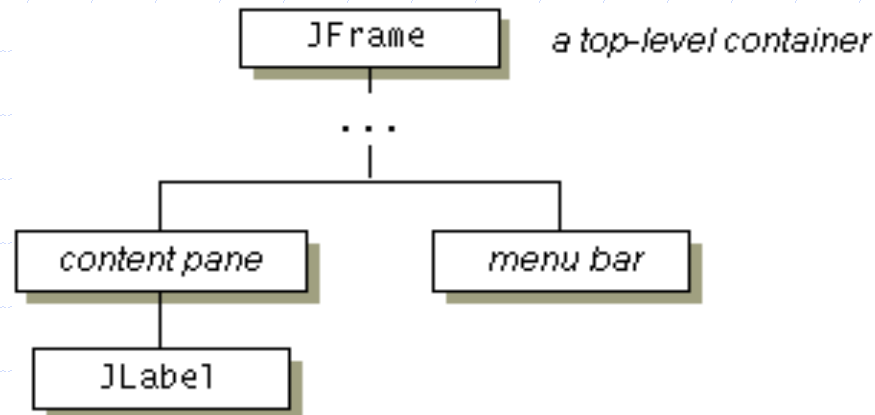
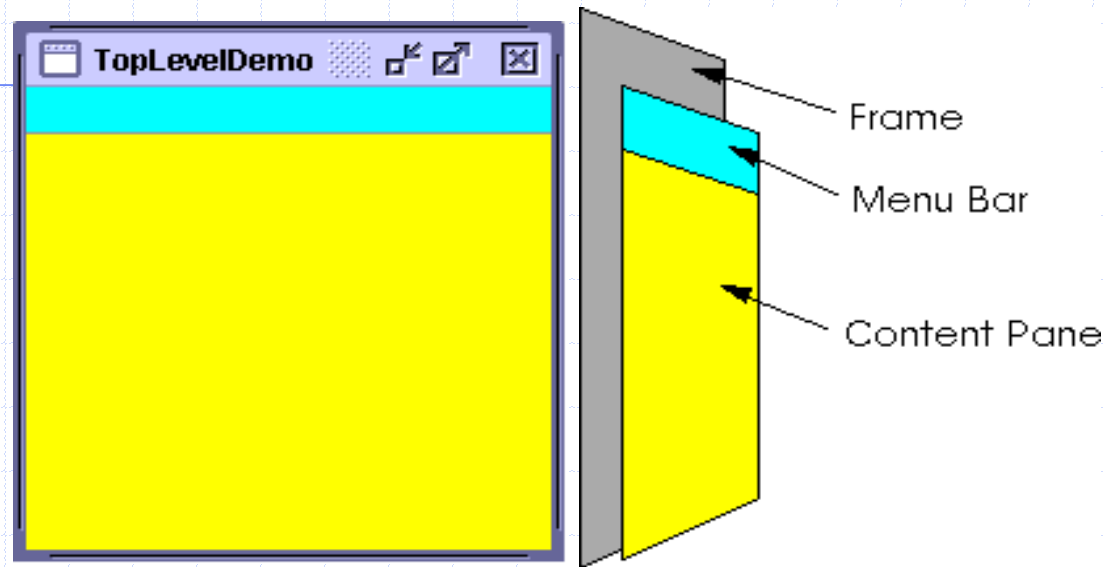


The Content Pane of a Frame

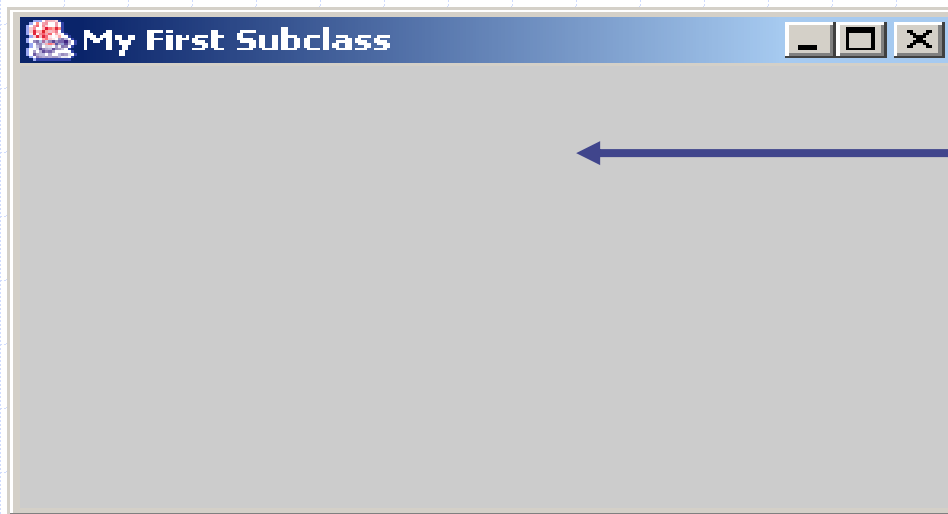


Use of Top Level Container



The Content Pane of a Frame

- The content pane is where we put GUI objects such as buttons, labels, scroll bars, and others.
- We access the content pane by calling the frame's **getContentPane** method.

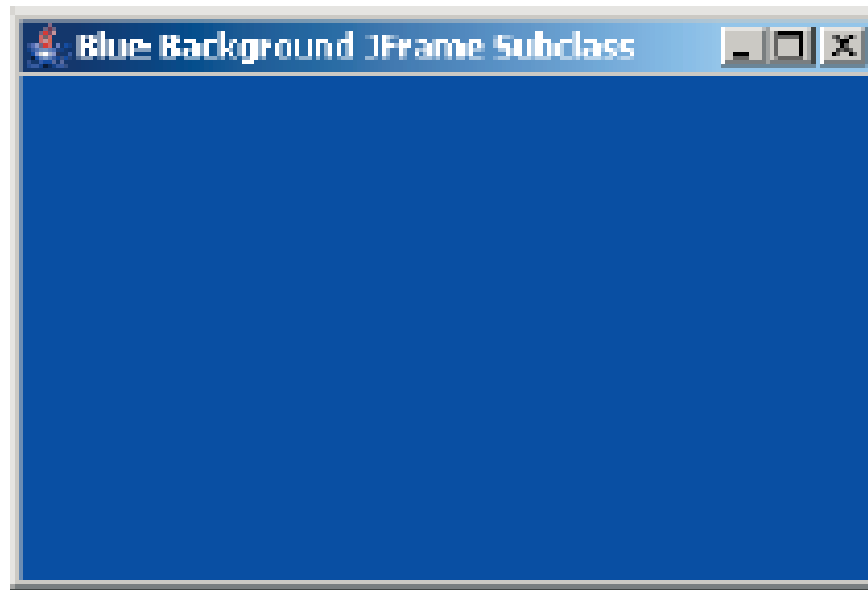


This gray area is the content pane of this frame.

Changing the Background Color

- Here's how we can change the background color of a content pane to blue:

```
Container contentPane = myFrame.getContentPane();  
contentPane.setBackground(Color.BLUE);
```



Adding Components to the Content Pane

```
myFrame.getContentPane().add(myLabel, BorderLayout.CENTER);
```

```
JPanel contentPane = new JPanel(new BorderLayout());  
contentPane.setBorder(someBorder);  
contentPane.add(someComponent, BorderLayout.CENTER);  
contentPane.add(anotherComponent, BorderLayout.PAGE_END);  
//Make it the content pane.  
contentPane.setOpaque(true);  
myFrame.setContentPane(contentPane);
```

Adding Components to the Content Pane

```
myFrame.getContentPane().add(myLabel, BorderLayout.CENTER);
```

```
JPanel contentPane = myFrame.getContentPane();  
contentPane.setBorder(someBorder);  
contentPane.add(someComponent, BorderLayout.CENTER);  
contentPane.add(anotherComponent, BorderLayout.PAGE_END);  
//Make it the content pane.  
contentPane.setOpaque(true);
```

Placing GUI Objects on a Frame

- There are two ways to put GUI objects on the content pane of a frame:
 - Use a *layout manager*
 - FlowLayout
 - BorderLayout
 - GridLayout
 - Use *absolute positioning*
 - null layout manager

JPanel API

Method

Purpose


[void add\(Component\)](#)
[void add\(Component, int\)](#)
[void add\(Component, Object\)](#)
[void add\(Component, Object, int\)](#)
[void add\(String, Component\)](#)

Add the specified component to the panel. When

present, the `int` parameter is the index of the component within the container. By default, the first component added is at index 0, the second is at index 1, and so on. The `Object` parameter is layout manager dependent and typically provides information to the layout manager regarding positioning and other layout constraints for the added component. The `String` parameter is similar to the `Object` parameter.

[int getComponentCount\(\)](#)

Get the number of components in this panel.

[Component](#)
[getComponent\(int\)](#)
[Component](#)
[getComponentAt\(int, int\)](#)
[Component](#)
[getComponentAt\(Point\)](#)
[Component\[\]](#)
[getComponents\(\)](#)

Get the specified component or components. You can get a component based on its index or *x, y* position.

[void remove\(Component\)](#)
[void remove\(int\)](#)
[void removeAll\(\)](#)

Remove the specified component(s).

Layout Managers

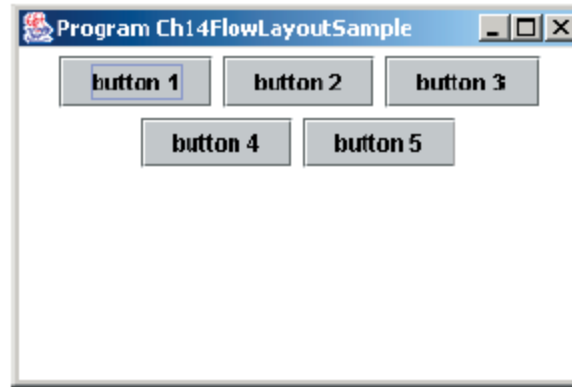
- The layout manager determines how the GUI components are added to the container (such as the content pane of a frame)
- Among the many different layout managers, the common ones are
 - FlowLayout
 - BorderLayout
 - GridLayout

FlowLayout

- In using this layout, GUI components are placed in left-to-right order.
 - When the component does not fit on the same line, left-to-right placement continues on the next line.
- As a default, components on each line are centered.
- When the frame containing the component is resized, the placement of components is adjusted accordingly.

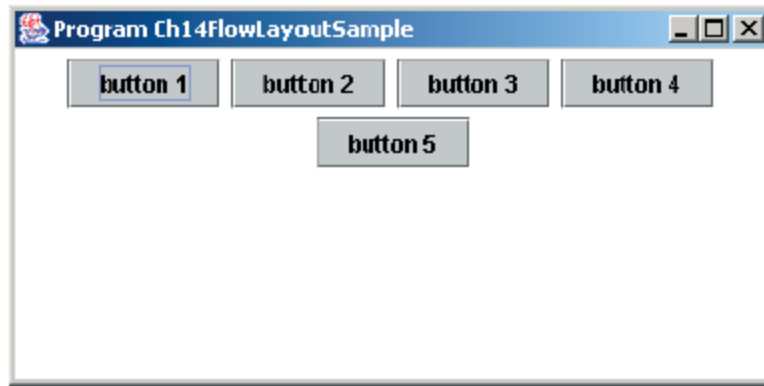
FlowLayout Sample

This shows the placement of five buttons by using FlowLayout.



When the frame first appears on the screen.

Center alignment is used as a default. It can be set to a different alignment at the time a FlowLayout is created.



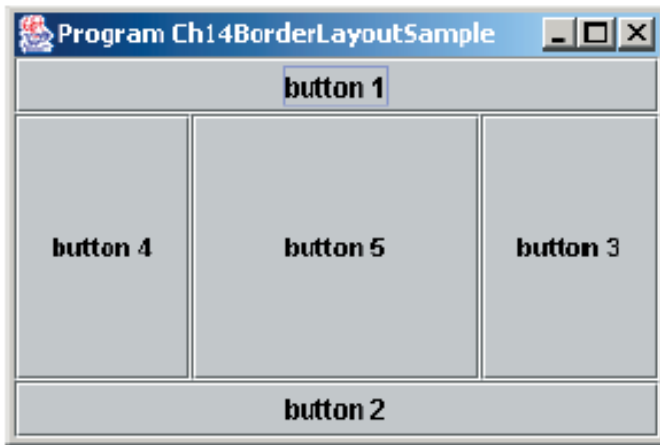
After the frame's width is widened and shortened.

BorderLayout

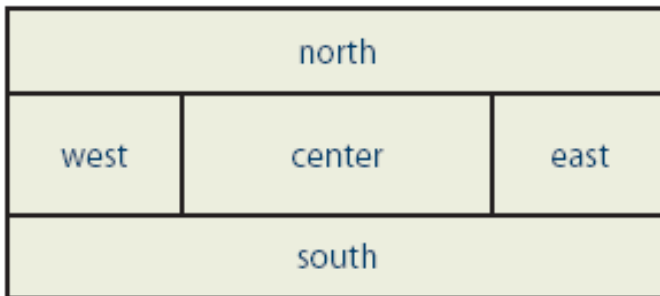
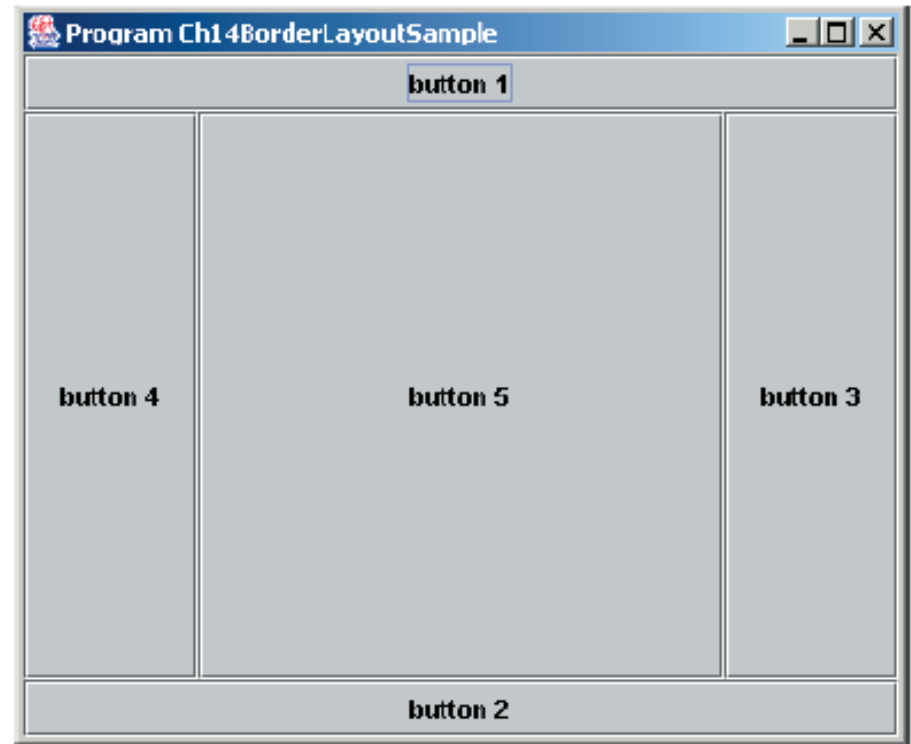
- This layout manager divides the container into five regions: center, north, south, east, and west.
 - The north and south regions expand or shrink in height only
 - The east and west regions expand or shrink in width only
 - The center region expands or shrinks on both height and width.
- 12 Not all regions have to be occupied.

BorderLayout Sample

When the frame first appears on the screen.



After the frame is resized.

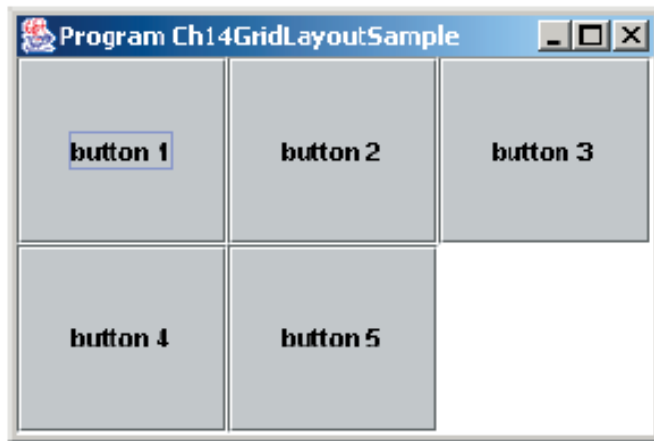


GridLayout

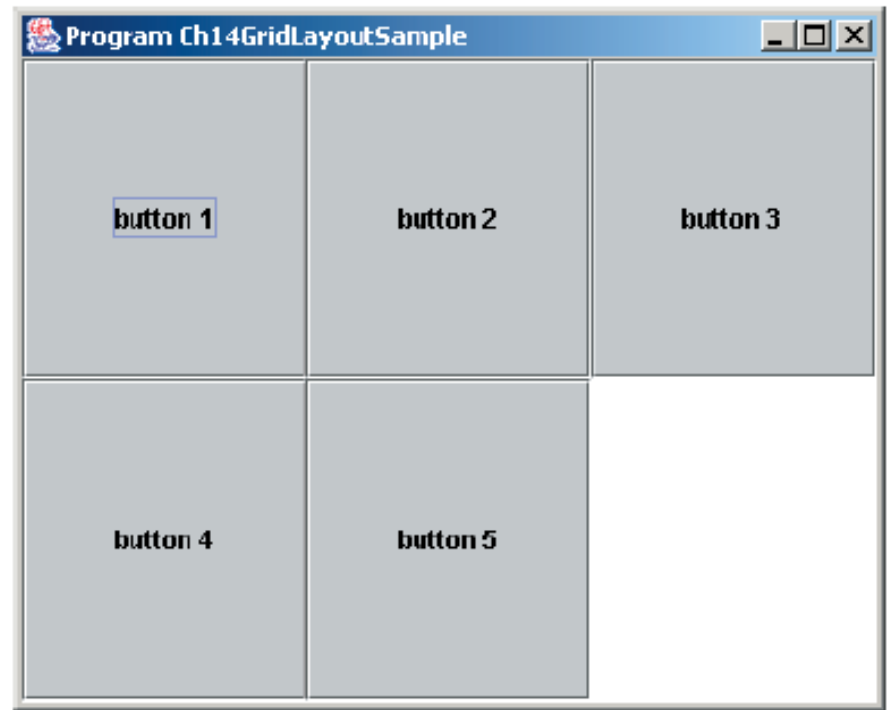
- This layout manager places GUI components on equal-size N by M grids.
- Components are placed in top-to-bottom, left-to-right order.
- The number of rows and columns remains the same after the frame is resized, but the width and height of each region will change.

GridLayout Sample

When the frame first appears on the screen.



After the frame is resized.



Nesting Panels

- It is possible, but very difficult, to place all GUI components on a single JPanel or other types of containers.
- A better approach is to use multiple panels, placing panels inside other panels.
- To illustrate this technique, we will create two sample frames that contain nested panels.