

# Chapter 14

## GUI and Event-Driven Programming



# 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` (see `Ch14FlowLayoutSample.java`)
  - `BorderLayout` (see `Ch14BorderLayoutSample.java`)
  - `GridLayout` (see `Ch14GridLayoutSample.java`)



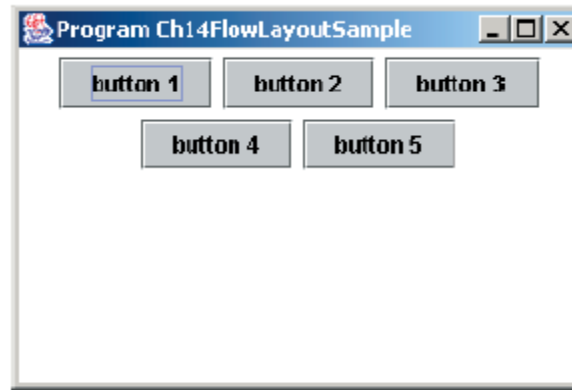
# FlowLayout

- 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.



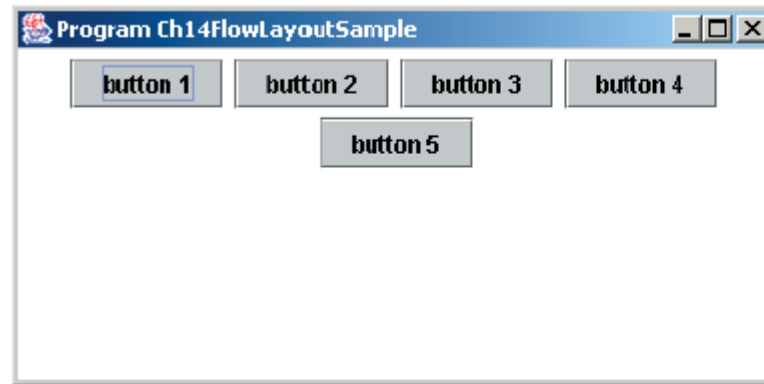
# FlowLayout Sample

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.



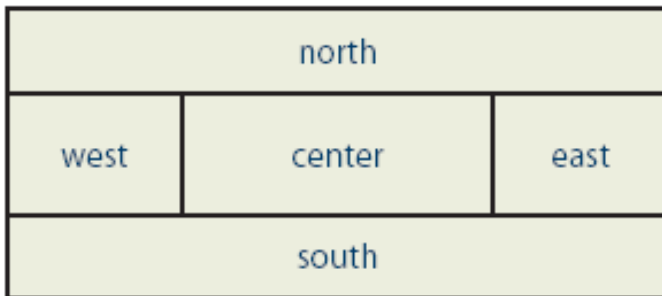
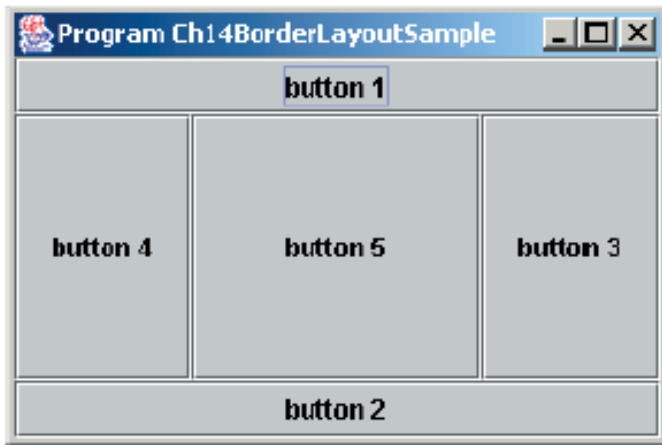
# BorderLayout

- 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.
- Not all regions have to be occupied.

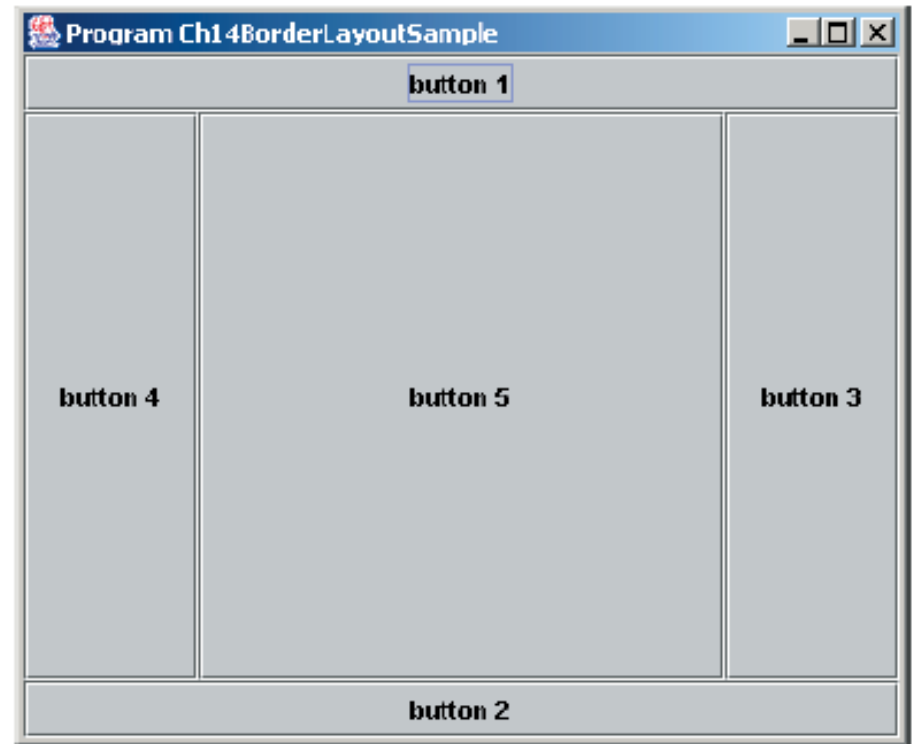


# BorderLayout Sample

When the frame first appears on the screen.



After the frame is resized.





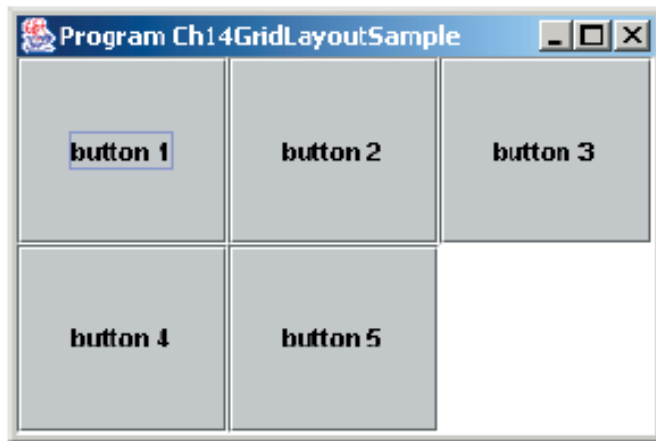
# GridLayout

- 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.

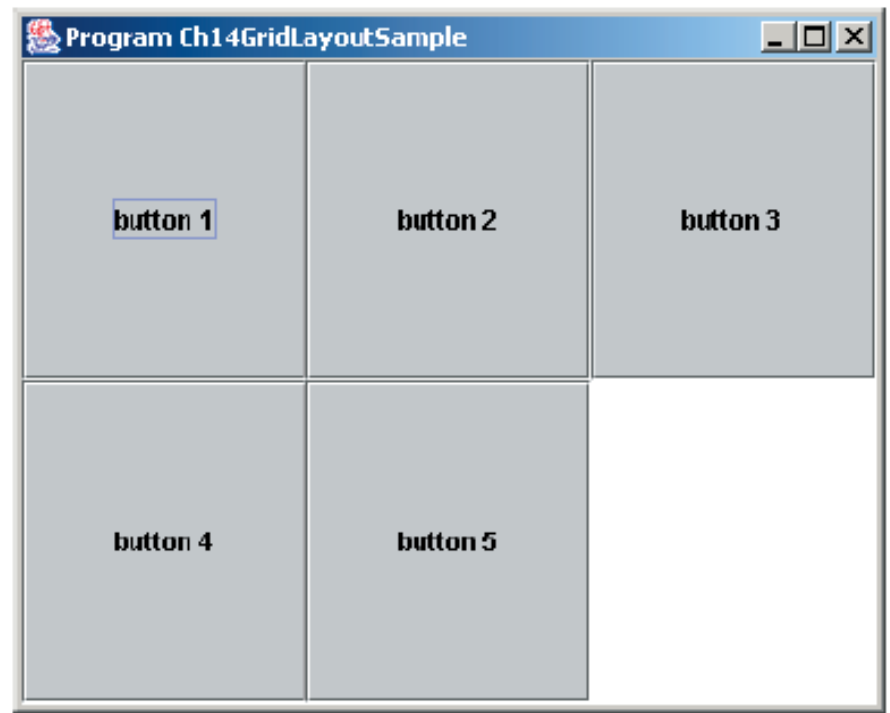


# GridLayout Sample

When the frame first appears on the screen.



After the frame is resized.







# Main Point

Components are arranged in a container through the use of *layout managers* that organize components in different ways. `FlowLayout` preserves the size of components and lays components out horizontally, from left to right. `BorderLayout` lays out components in five positions – north, south, east, west and center; to preserve the size of components, `BorderLayout` is used in conjunction with `FlowLayout`. Likewise, all of manifest life is conducted by a vast network of natural laws.



# 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.
- Ch14NestedPanels1.java provides the user interface for playing Tic Tac Toe.
- Ch14NestedPanels2.java provides the user interface for playing HiLo.



## Main Point

Because containers are themselves a certain type of component, containers can be organized inside of other containers. Attractive visual design of GUIs is accomplished in Swing through the creative use of multiple layouts of container classes. The natural order of existence is created and maintained by the hidden dynamics of pure intelligence.



# Other Common GUI Components

- **JCheckBox**
  - see Ch14JCheckBoxSample1.java and Ch14JCheckBoxSample2.java
- **JRadioButton**
  - see Ch14JRadioButtonSample.java
- **JComboBox**
  - see Ch14JComboBoxSample.java
- **JList**
  - see Ch14JListSample.java
- **JSlider**
  - see Ch14JSliderSample.java

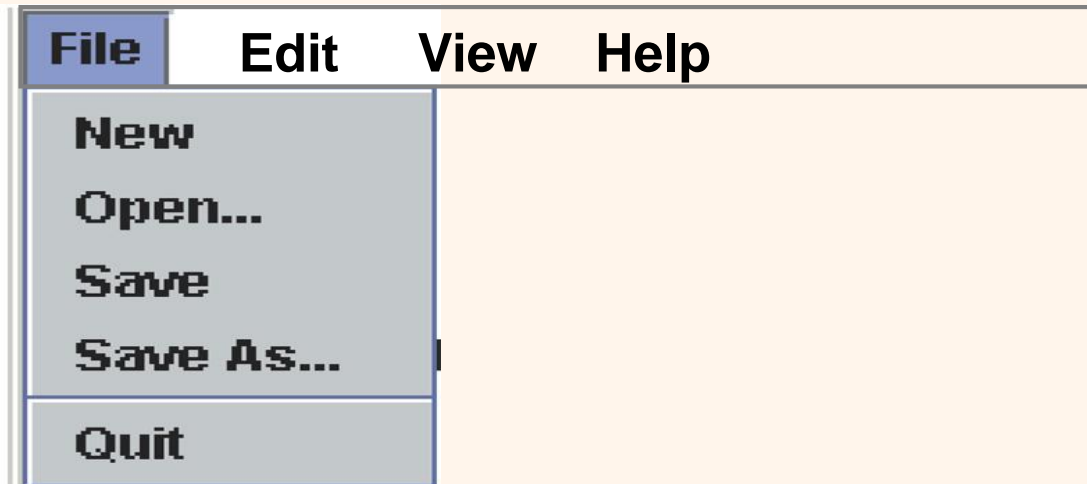


# Menus

- The javax.swing package contains three menu-related classes: **JMenuBar**, **JMenu**, and **JMenuItem**.
- JMenuBar is a bar where the menus are placed. There is one menu bar per frame.
- JMenu (such as File or Edit) is a group of menu choices. JMenuBar may include many JMenu objects.
- JMenuItem (such as Copy, Cut, or Paste) is an individual menu choice in a JMenu object.
- Only the JMenuItem objects generate events.



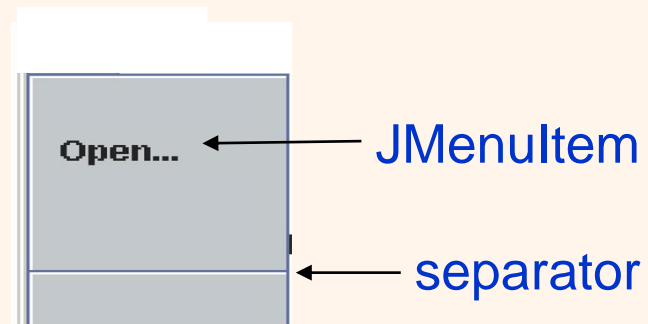
# Menu Components



JMenuBar



JMenu





# Sequence for Creating Menus

1. Create a JMenuBar object and attach it to a frame.
2. Create a JMenu object.
3. Create JMenuItem objects and add them to the JMenu object.
4. Attach the JMenu object to the JMenuBar object.



# Handling Mouse Events

- Mouse events include such user interactions as
  - moving the mouse
  - dragging the mouse (moving the mouse while the mouse button is being pressed)
  - clicking the mouse buttons.
- The `MouseListener` interface handles mouse button
  - `mouseClicked`, `mouseEntered`, `mouseExited`, `mousePressed`, and `mouseReleased`
- The `MouseMotionListener` interface handles mouse movement
  - `mouseDragged` and `mouseMoved`.
- See `Ch14TrackMouseFrame` and `Ch14SketchPad`





# Connecting the Parts of Knowledge With the Wholeness of Knowledge

*The self-referral dynamics  
arising from the reflexive association of container classes*

1. In Swing, components are placed and arranged in container classes for attractive display.
  2. In Swing, containers are also considered to be components; this makes it possible to place and arrange container classes inside other container classes. These self-referral dynamics support a much broader range of possibilities in the design of GUIs.
- 
3. Transcendental Consciousness: TC is the self-referral field of all possibilities.
  4. Wholeness moving within itself: In Unity Consciousness, all activity is appreciated as the self-referral dynamics of one's own Self.

