Inner Class

• An inner class is a class defined within another class.
• An inner object is bonded with its external object.
• The inner object has access to all (both public and private) the variables and methods of the external object.
Inner object as event listener

• Rewrite TwoButtons.java class and rename it to TwoButtonsInnerClass.java
• Since we have two buttons, we will create two inner class object, each serves as a event listener for only one button.
• We will register each button with one inner class object.
Layout Manager

• FlowLayout
• BorderLayout
• GridLayout
FlowLayout

• Place components sequentially (left to right) in the order they were added.

• When the edge of the container is reached, components continue to display on the next line.
FlowLayoutDemo.java

• Data: three buttons, one layout manager, one container.
• Two methods: go(), and actionPerformed()
Data

• There are three buttons,
  JButton left JButton;
  JButton center JButton;
  JButton right JButton;
• There is one flow layout manager
  FlowLayout layout;
• There is one container.
  Container container;
Methods

- go() method that creates the frame, create buttons, set the layout, add the buttons to the content pane on the frame, set the pane size, add action listener, set the pane visible.
- actionPerformed() method is part of implementation of ActionListener interface.
BorderLayout

- BorderLayout layout manager arranges components into five regions: north, south, center, east, west.
- BorderLayout is the default layout manager for JFrame.
Demo BorderLayout
GridLayout

• GridLayout layout manager divides the container into a grid so that components can be placed in rows and columns.
• Components are added to a GridLayout starting at the top-left cell of the grid and proceeding left to right until the row is full. Then the process continues left to right on the next row of the grid.
Demo GridLayout