**OOP LECTURE 15-16**

**Layout Manager**

Layout refers to the arrangement of components within the container. In another way, it could be said that layout is placing the components at a particular position within the container. The task of laying out the controls is done automatically by the Layout Manager.

The layout manager automatically positions all the components within the container. Even if you do not use the layout manager, the components are still positioned by the default layout manager. It is possible to lay out the controls by hand, however, it becomes very difficult because of the following two reasons.

* It is very tedious to handle a large number of controls within the container.
* Usually, the width and height information of a component is not given when we need to arrange them.

Java provides various layout managers to position the controls. Properties like size, shape, and arrangement varies from one layout manager to the other. When the size of the applet or the application window changes, the size, shape, and arrangement of the components also changes in response, i.e. the layout managers adapt to the dimensions of the appletviewer or the application window.

The layout manager is associated with every Container object. Each layout manager is an object of the class that implements the LayoutManager interface.

**Types of Layouts**

In Java GUI Swing, there are several types of layouts you can use to organize components within a container. Each layout manager has its own way of arranging components. Here are some common layouts:

1. [**NullLayout:**](https://www.youtube.com/watch?v=0pC5T4hMn2I&list=PLjJmj2FyqToa6ktTILKN-nEXsqn2k8F6t&index=2)Null layout, sometimes referred to as absolute positioning, is a layout manager in Java Swing that allows you to explicitly specify the position and size of each component within a container. Unlike other layout managers that automatically arrange components based on certain rules or constraints, null layout gives you complete control over the placement of components.
2. [**FlowLayout**:](https://www.youtube.com/watch?v=217CFPR5LD8&list=PLjJmj2FyqToa6ktTILKN-nEXsqn2k8F6t&index=4) Components are arranged in a row, one after another. If the row is filled, the next component is placed on the next row.
3. [**BorderLayout**:](https://www.youtube.com/watch?v=28wX3W2C22s&list=PLjJmj2FyqToa6ktTILKN-nEXsqn2k8F6t&index=3) Components are arranged in five regions: North, South, East, West, and Center. Each region can contain only one component.
4. [**GridLayout**:](https://www.youtube.com/watch?v=sAjL8C3XC0U&list=PLjJmj2FyqToa6ktTILKN-nEXsqn2k8F6t&index=5) Components are arranged in a grid of rows and columns. All components are of the same size.
5. [**BoxLayout**](https://www.youtube.com/watch?v=U2riAdKz_xA&list=PLjJmj2FyqToa6ktTILKN-nEXsqn2k8F6t&index=9): Components are arranged either horizontally or vertically in a single line.
6. **CardLayout**: Components are stacked on top of each other, and only one component is visible at a time. You can switch between components like flipping through cards.
7. **GridBagLayout**: This is the most flexible layout manager. It arranges components in a grid, but each component can occupy multiple cells and have different sizes.
8. **GroupLayout**: Introduced in Java SE 6, GroupLayout is a layout manager that groups components and sizes them dynamically. It's often used in GUI builders like NetBeans.

**NULL LAYOUT**

**NullLayout Sample Example:**

import javax.swing.\*;

import java.awt.\*;

public class NullLayoutExample {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("Null Layout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(300, 200);

// Create a JPanel with null layout

JPanel panel = new JPanel();

panel.setLayout(null);

// Create components and set their bounds explicitly

JLabel label1 = new JLabel("Username:");

label1.setBounds(20, 20, 80, 25);

JTextField textField = new JTextField();

textField.setBounds(100, 20, 160, 25);

JLabel label2 = new JLabel("Password:");

label2.setBounds(20, 60, 80, 25);

JPasswordField passwordField = new JPasswordField();

passwordField.setBounds(100, 60, 160, 25);

JButton loginButton = new JButton("Login");

loginButton.setBounds(100, 100, 80, 30);

// Add components to the panel

panel.add(label1);

panel.add(textField);

panel.add(label2);

panel.add(passwordField);

panel.add(loginButton);

// Add panel to the frame

frame.add(panel);

// Set frame visibility

frame.setVisible(true);

}

}

**NullLayout Sample Example without panel**

**2.**

import javax.swing.\*;

import java.awt.\*;

public class Main{

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("Null Layout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(300, 200);

JLabel label1 = new JLabel("Username:");

label1.setBounds(20, 20, 80, 25);

JTextField textField = new JTextField();

textField.setBounds(100, 20, 160, 25);

JLabel label2 = new JLabel("Password:");

label2.setBounds(20, 60, 80, 25);

JPasswordField passwordField = new JPasswordField();

passwordField.setBounds(100, 60, 160, 25);

JButton loginButton = new JButton("Login");

loginButton.setBounds(100, 100, 80, 30);

frame.add(label1);

frame.add(textField);

frame.add(label2);

frame.add(passwordField);

frame.add(loginButton);

frame.setLayout(null);

frame.setVisible(true);

}

}

**FLOW LAYOUT**

**FlowLayout Sample Example:**

import javax.swing.\*;

import java.awt.\*;

public class FlowLayoutExample {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("FlowLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create a JPanel with FlowLayout

JPanel panel = new JPanel(new FlowLayout());

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

JButton button4 = new JButton("Button 4");

JButton button5 = new JButton("Button 5");

// Add components to the panel

panel.add(button1);

panel.add(button2);

panel.add(button3);

panel.add(button4);

panel.add(button5);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

frame.setSize(300, 150);

frame.setVisible(true);

}

}

**FlowLayout Example Without JPanel:**

import javax.swing.\*;

import java.awt.\*;

public class FlowLayoutExample {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("FlowLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

JButton button4 = new JButton("Button 4");

JButton button5 = new JButton("Button 5");

// Set layout manager for the JFrame

frame.setLayout(new FlowLayout());

// Add components to the frame

frame.add(button1);

frame.add(button2);

frame.add(button3);

frame.add(button4);

frame.add(button5);

// Set frame size and visibility

frame.setSize(300, 150);

frame.setVisible(true);

}

}

**FlowLayout Sample Example with alignment on left**

import javax.swing.\*;

import java.awt.\*;

public class Main1 {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("FlowLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create a JPanel with FlowLayout

JPanel panel = new JPanel(new FlowLayout(FlowLayout.LEFT));

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

JButton button4 = new JButton("Button 4");

JButton button5 = new JButton("Button 5");

// Add components to the panel

panel.add(button1);

panel.add(button2);

panel.add(button3);

panel.add(button4);

panel.add(button5);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

frame.setSize(300, 150);

frame.setVisible(true);

}

}

**FlowLayout Sample example with alignment on Right:**

import javax.swing.\*;

import java.awt.\*;

public class Main1 {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("FlowLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create a JPanel with FlowLayout

JPanel panel = new JPanel(new FlowLayout(FlowLayout.RIGHT));

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

JButton button4 = new JButton("Button 4");

JButton button5 = new JButton("Button 5");

// Add components to the panel

panel.add(button1);

panel.add(button2);

panel.add(button3);

panel.add(button4);

panel.add(button5);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

frame.setSize(300, 150);

frame.setVisible(true);

}

}

**FlowLayout Horizontal and Vertical Gaps:**  
In Java Swing, the FlowLayout class provides methods to control the horizontal and vertical gaps between components. However, setHGap is not a direct method in FlowLayout. Instead, you typically set the horizontal and vertical gaps using the constructor or by using FlowLayout's constructor that accepts gap parameters.

Here's an example demonstrating how to set the horizontal gap using FlowLayout constructor:

import javax.swing.\*;

import java.awt.\*;

public class FlowLayoutExample {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("FlowLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create a JPanel with FlowLayout and set horizontal gap

JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));

// Here, 20 is the horizontal gap, and 10 is the vertical gap

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

// Add components to the panel

panel.add(button1);

panel.add(button2);

panel.add(button3);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

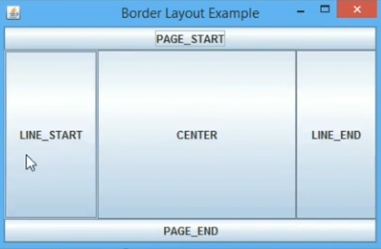
frame.setSize(300, 150);

frame.setVisible(true);

}

}

**BORDER LAYOUT**



**BorderLayout Sample Example:**



import javax.swing.\*;

import java.awt.\*;

public class BorderLayoutExample {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("BorderLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create components

JButton northButton = new JButton("North");

JButton southButton = new JButton("South");

JButton eastButton = new JButton("East");

JButton westButton = new JButton("West");

JButton centerButton = new JButton("Center");

// Create a JPanel with BorderLayout

JPanel panel = new JPanel(new BorderLayout());

// Add components to the panel with specific regions

panel.add(northButton, BorderLayout.NORTH);

panel.add(southButton, BorderLayout.SOUTH);

panel.add(eastButton, BorderLayout.EAST);

panel.add(westButton, BorderLayout.WEST);

panel.add(centerButton, BorderLayout.CENTER);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

frame.setSize(300, 200);

frame.setVisible(true);

}

}

**Adding multiple components in border layout:**

Yes, you can add multiple components to the same region (north, south, east, west) in a BorderLayout. When you add multiple components to the same region, they are stacked vertically (for north and south) or horizontally (for east and west) in the order they are added.

Here's how you can add multiple items to the north, south, east, and west regions in BorderLayout:



import javax.swing.\*;

import java.awt.\*;

public class Main1 {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("BorderLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create components

JButton northButton1 = new JButton("North Button 1");

JButton northButton2 = new JButton("North Button 2");

JButton southButton1 = new JButton("South Button 1");

JButton southButton2 = new JButton("South Button 2");

JButton eastButton1 = new JButton("East Button 1");

JButton eastButton2 = new JButton("East Button 2");

JButton westButton1 = new JButton("West Button 1");

JButton westButton2 = new JButton("West Button 2");

// Create a JPanel with BorderLayout

JPanel mainPanel = new JPanel(new BorderLayout());

// Create a JPanel for each region with GridLayout(1, 2)

JPanel northPanel = new JPanel(new GridLayout(1, 2));

JPanel southPanel = new JPanel(new GridLayout(1, 2));

JPanel eastPanel = new JPanel(new GridLayout(2, 1));

JPanel westPanel = new JPanel(new GridLayout(2, 1));

// Add buttons to the panels

northPanel.add(northButton1);

northPanel.add(northButton2);

southPanel.add(southButton1);

southPanel.add(southButton2);

eastPanel.add(eastButton1);

eastPanel.add(eastButton2);

westPanel.add(westButton1);

westPanel.add(westButton2);

// Set layout for each region

mainPanel.add(northPanel, BorderLayout.NORTH);

mainPanel.add(southPanel, BorderLayout.SOUTH);

mainPanel.add(eastPanel, BorderLayout.EAST);

mainPanel.add(westPanel, BorderLayout.WEST);

// Add main panel to the frame

frame.add(mainPanel);

// Set frame size and visibility

frame.setSize(400, 300);

frame.setVisible(true);

}

}

**Grid Layout:**

**10.**

import javax.swing.\*;

import java.awt.\*;

public class Main1 {

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("GridLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

JButton button4 = new JButton("Button 4");

JButton button5 = new JButton("Button 5");

// Create a JPanel with GridLayout

JPanel panel = new JPanel(new GridLayout(2, 3)); // 2 rows, 3 columns

// Add components to the panel

panel.add(button1);

panel.add(button2);

panel.add(button3);

panel.add(button4);

panel.add(button5);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

frame.setSize(300, 200);

frame.setVisible(true);

}

}

**BOX LAYOUT**

**11.**

import javax.swing.\*;

import java.awt.\*;

public class Main1{

public static void main(String[] args) {

// Create a JFrame

JFrame frame = new JFrame("BoxLayout Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create components

JButton button1 = new JButton("Button 1");

JButton button2 = new JButton("Button 2");

JButton button3 = new JButton("Button 3");

// Create a JPanel with BoxLayout (vertical)

JPanel panel = new JPanel();

//panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS)); // Arrange components vertically

panel.setLayout(new BoxLayout(panel, BoxLayout.X\_AXIS));

// Add components to the panel

panel.add(button1);

panel.add(button2);

panel.add(button3);

// Add panel to the frame

frame.add(panel);

// Set frame size and visibility

frame.setSize(200, 200);

frame.setVisible(true);

}

}