**What is a Container Class?**

Container classes are classes that can have other components on it. So for creating a Java Swing GUI, we need at least one container object.

There are 3 types of Java Swing containers.

1. **Panel**: It is a pure container and is not a window in itself. The sole purpose of a Panel is to organize the components on to a window.
2. **Frame**: It is a fully functioning window with its title and icons.
3. **Dialog**: It can be thought of like a pop-up window that pops out when a message has to be displayed. It is not a fully functioning window like the Frame.

**What is GUI in Java?**

**GUI (Graphical User Interface) in Java** is an easy-to-use visual experience builder for Java applications. It is mainly made of graphical components like buttons, labels, windows, etc. through which the user can interact with an application. GUI plays an important role to build easy interfaces for Java applications.

**How to Make a GUI in Java with Example**

Now in this Java GUI Tutorial, let’s understand how to create a GUI in Java with Swings in Java examples.

**Step 1)** Copy code into an editor  
In first step Copy the following code into an editor.

import javax.swing.\*;

class gui

{

public static void main(String args[])

{

JFrame **frame** = new JFrame("My First GUI");

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

**frame**.setSize(300,300);

JButton button1 = new JButton("Press");

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

**frame**.getContentPane().add(button1); // Adds Button

**frame**.getContentPane().add(button2); // Adds Button

**frame**.setVisible(true);

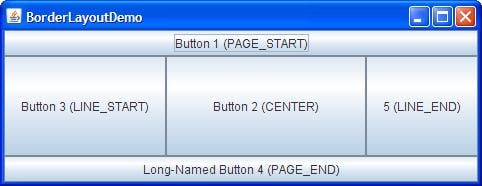
}

}

## Layout Manager

## BorderLayout

A BorderLayout places components in up to five areas: top, bottom, left, right, and center. It is the default layout manager for every java JFrame



## FlowLayout

FlowLayout is the default layout manager for every JPanel. It simply lays out components in a single row one after the other.

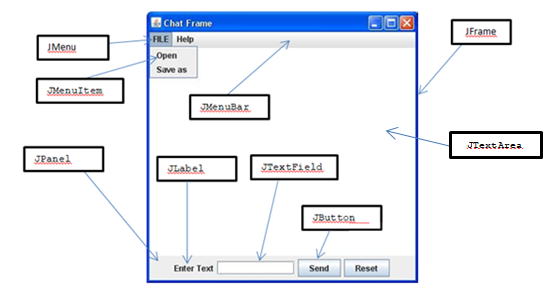


## Java GridBagLayout

It is the more sophisticated of all layouts. It aligns components by placing them within a grid of cells, allowing components to span more than one cell.



**Step 8)** Create chat frame  
How about creating a chat frame like below?



//Usually you will require both swing and **awt** packages

//even if you are working with just swings.

import javax.swing.\*;

import java.awt.\*;

class gui

{

public static void main(String args[])

{

**//Creating the Frame**

JFrame **frame** = new JFrame("Chat Frame");

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

**frame**.setSize(400, 400);

**//Creating the MenuBar and adding components**

JMenuBar **menubarObj** = new JMenuBar();

JMenu **menu1** = new JMenu("FILE");

JMenu **menu2** = new JMenu("Help");

**menubarObj**.add(**menu1**);

**menubarObj**.add(**menu2**);

JMenuItem **menuitem1** = new JMenuItem("Open");

JMenuItem **menuitem2** = new JMenuItem("Save as");

**menu1**.add(**menuitem1**);

**menu1**.add(**menuitem2**);

**//Creating the panel at bottom and adding components**

JPanel panel = new JPanel(); // the panel is not visible in output

JLabel **label1** = new JLabel("Enter Text");

JTextField **textfield1** = new JTextField(10); // accepts upto 10 characters

JButton **sendbtn** = new JButton("Send");

JButton **resetbtn** = new JButton("Reset");

panel.add(**label1**); // Components Added using Flow Layout

panel.add(**textfield1**);

panel.add(**sendbtn**);

panel.add(**resetbtn**);

**// Text Area at the Center**

JTextArea **textarea1** = new JTextArea();

**//Adding Components to the frame.**

**frame**.getContentPane().add(BorderLayout.NORTH, **menubarObj**);

**frame**.getContentPane().add(BorderLayout.CENTER, **textarea1**);

**frame**.getContentPane().add(BorderLayout.SOUTH, panel);

**frame**.setVisible(true);

}

}