Java Foundation class

The java foundation classes enable the original AWT by adding a set of GUI class libraries. JFC provide us with additional visual component classes and a unique way of designing the screen.

Swing

Swing is a set of classes under the JFC that provide lightweight visual components and enable creation of an attractive GUI. Swing not only contains replacement components for AWT visual components but also complex components like trees and tables that do not have AWT equivalents. Swing components have a pluggable look and feel so that with a single set of components we can achieve the look and feel of any OS platform.

In Swing, the main window, also called a top-level container, is the root of a hierarchy, which contains all of the Swing components that appear inside the window. All Swing applications have at least one top-level container.

Every top-level container has an intermediate container called content pane. This content pane contains all of the visible components in the GUI window. The content pane is the base pane upon which all other component or container objects are placed. One exception to this rule is, if there is a menu bar in the top-level container. Tis menu bar will have a special place, in the top -level container, which is outside the content pane.

All Swing components names start with J .For instance, the Swing button class is named JButton ,whereas the AWT button class is named Button. The Swing components are in the javax.swing package.

No.	Java AWT	Java Swing
1)	AWT components are **platform-dependent.**	Java swing components are **platform-independent.**
2)	AWT components are **heavyweight.**	Swing components are **lightweight.**
3)	AWT **doesn't support pluggable look and feel.**	Swing **supports pluggable look and feel.**
4)	AWT provides **less components** than Swing.	Swing provides **more powerful components** such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
5)	AWT **doesn't follows MVC**(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.	Swing **follows MVC.**

Top level containers in Swing are-

- Frames
- Dialogs
- Applets

The methods of Component class are widely used in java swing that are given below.

Method	Description
public void add(Component c)	add a component on another component.
public void setSize(int width,int height)	sets size of the component.
public void setLayout(LayoutManager m)	sets the layout manager for the component.
public void setVisible(boolean b)	sets the visibility of the component. It is by default false.

First Sample Program

```
import javax.swing.*;
public class jsampe
{
public static void main(String[] args)
{
JFrame f=new JFrame();                  //creating instance of JFrame

JButton b=new JButton("click Me");      //creating instance of JButton
b.setBounds(130,100,100, 40);           //x axis, y axis, width, height

f.add(b);                               //adding button in JFrame

f.setSize(400,500);                     //400 width and 500 height
f.setLayout(null);                      //using no layout managers
f.setVisible(true);                     //making the frame visible
}
}
```

JFrame

The JFrame is a top-level container or window, which provides a place for placing other Swing components.
A JFrame components is used to create windows in a swing program.

```
import java.awt.*;
import javax.swing.*;

public class Jfa extends JFrame
{
```

```
	public Jfa(String title)
	{
		super(title);
	}
	public static void main (String args[])
	{
		Jfa j=new Jfa("Using Swing");
		j.setVisible(true);
		j.setSize(200,200);
	}
}
```

JLabel ---

```
import java.awt.*;
import javax.swing.*;

public class Jlab extends JApplet
{
	public void init()
	{
		getContentPane().setLayout(new FlowLayout());
		ImageIcon ic=new ImageIcon("a.gif");
		JLabel l1=new JLabel(ic);
		getContentPane().add(l1);
	}
}
```

JTextField-- The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text.

```
import java.awt.*;
import javax.swing.*;

public class Jtext extends JFrame
{
	public Jtext()
	{
		Container con=getContentPane();
		con.setLayout(new FlowLayout());
		JLabel l1=new JLabel("Enter ypur name ");
		JTextField t1=new JTextField(20);
		con.add(l1);
		con.add(t1);
		setVisible(true);
		setSize(200,200);

	}
	public static void main(String args[])
	{
```

```
                Jtext j=new Jtext();
        }
}
```

JButton-- The JButton class is used to create a labeled button that has platform independent implementation.
[We can use **setBounds**(x, y, width, height) to specify the position and size of a GUI component if you set the layout to null . Then (x, y) is the coordinate of the upper-left corner of that component. **setBounds** is used to define the bounding rectangle of a component. This includes it's position and size.]

```
import javax.swing.*;
public class Jb
{
public static void main(String[] args)
{
   JFrame f=new JFrame("Button");
   JButton b=new JButton("Click Here");
   b.setBounds(50,100,95,30);
   f.add(b);
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
}
}
```

Example 2 [Using No Layout]

```
import javax.swing.*;
public class Jb
{
public static void main(String[] args)
{
   JFrame f=new JFrame("Button");
   JLabel l1=new JLabel("Enter your name ");
   l1.setBounds(10,10,100,30);
   JTextField t1=new JTextField(20);
   t1.setBounds(120,10,100,30);
   JButton b=new JButton("Click Here");
   b.setBounds(20,50,95,30);
   f.add(l1);
   f.add(t1);
   f.add(b);
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
```

```
}
}
```

Example 3: Using Flowlayout

```
import java.awt.*;
import javax.swing.*;
public class jb2
{
public static void main(String[] args)
{
   JFrame f=new JFrame("Button");
   JLabel l1=new JLabel("Enter your name ");
   JTextField t1=new JTextField(20);
   JButton b=new JButton("Click Here");
   f.add(l1);
   f.add(t1);
   f.add(b);
   f.setSize(400,400);
   f.setLayout(new FlowLayout(FlowLayout.LEFT));
   f.setVisible(true);   } }
```

Example 4

```
import java.awt.*;
import javax.swing.*;
public class jlab3
{
public static void main(String[] args)
{
   JFrame f=new JFrame("Enter your 1st No :");
   JLabel l1=new JLabel("Enter your name ");

   JTextField t1=new JTextField(20);
   JLabel l2=new JLabel("Enter your 2nd no :");

   JTextField t2=new JTextField(20);
   JLabel l3=new JLabel(" Result ");

   JTextField t3=new JTextField(20);

   JButton b=new JButton("OK");
   f.add(l1);
   f.add(t1);
   f.add(l2);
   f.add(t2);
   f.add(l3);
   f.add(t3);
```

```
    f.add(b);
    f.setSize(400,400);
    f.setLayout(new FlowLayout(FlowLayout.LEFT));
    f.setVisible(true);
}
}
```

Example 5 [Find Factorial value in swing (ActionEvent is used)]

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class jlab3 implements ActionListener
{
JLabel l1,l2;
JTextField t1,t2;
JButton b1;
public jlab3()
{

    JFrame f=new JFrame("Swing Frame :");
    l1=new JLabel("Enter your no :");

    t1=new JTextField(10);

    l2=new JLabel(" Result ");

    t2=new JTextField(10);

    b1=new JButton("OK");
    b1.addActionListener(this);
    f.add(l1);
    f.add(t1);

    f.add(l2);
    f.add(t2);
    f.add(b1);
    f.setSize(400,100);
    f.setLayout(new FlowLayout(FlowLayout.LEFT));
    f.setVisible(true);
}

public void actionPerformed(ActionEvent e)
{
if(e.getSource( )==b1)
{
```

```
int n = Integer.parseInt (t1.getText ( ));
int m=1,i;
for(i=1;i<=n;i++)
{
        m=m*i;
}

t2.setText (String.valueOf (m));
}
}
public static void main(String args[])
{
        jlab3 jl=new jlab3();
}
}
```

JTextArea- The object of a JTextArea class is a multiline region that displays text. It allows the editing of multiple line text.

```
import javax.swing.*;
public class jtextarea
{
    jtextarea()
    {
      JFrame f= new JFrame();
      JTextArea area=new JTextArea("Welcome to java swing programming");
      area.setBounds(10,30, 200,200);
      f.add(area);
      f.setSize(300,300);
      f.setLayout(null);
      f.setVisible(true);
    }
public static void main(String args[])
   {
  new jtextarea();
   }}
```