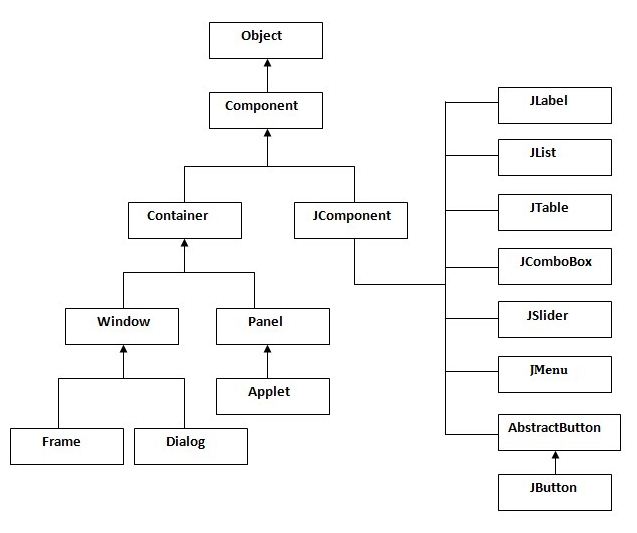
**Swing API** is a set of extensible GUI Components to ease to create JAVA based Front End/GUI Applications. It is build on top of AWT API and acts as a replacement of AWT API. Java Swing is a part of Java Foundation Classes (JFC).Java Swing provides platform-independent and lightweight components. The main package is **javax.swing.**The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Swing component follows a **Model-View-Controller(MVC)** architecture.

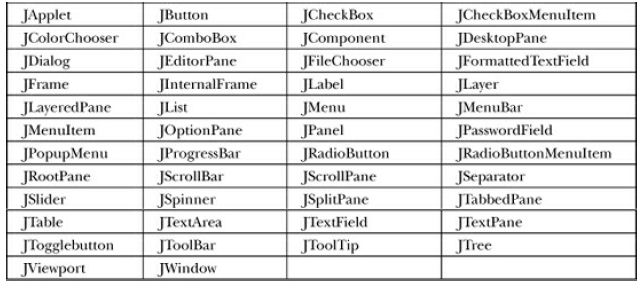
**Components and Containers**

A Swing GUI consists of two key items: components and containers. However, this distinction is mostly conceptual because all containers are also components. The difference between the two is found in their intended purpose: As the term is commonly used, a component is an independent visual control, such as a push button or slider. A container holds a group of components. Thus, a container is a special type of component that is designed to hold other components.

**HIERARCHY OF SWING**



**The Swing Packages**



**Simple Java Swing Example**

import javax.swing.\*;

public class Swing1 {

public static void main(String[] args) {

JFrame f=new JFrame();

JButton b=new JButton("click");

b.setBounds(130,100,100, 40);

f.add(b);

f.setSize(400,500);

f.setLayout(null);

f.setVisible(true);

}

}

===================================================================================

**Java JButton Example**

import javax.swing.\*;

public class ButtonExample {

public static void main(String[] args) {

JFrame f=new JFrame("Button Example");

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 of displaying image on the button:**

import javax.swing.\*;

public class ButtonExample{

ButtonExample(){

JFrame f=new JFrame("Button Example");

JButton b=new JButton(new ImageIcon("D:\\icon.png"));

b.setBounds(100,100,100, 40);

f.add(b);

f.setSize(300,400);

f.setLayout(null);

f.setVisible(true);

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

}

public static void main(String[] args) {

new ButtonExample();

}

}

====================================================================================

**Java swing JLabel**

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. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

**JLabel class declaration**

public class JLabel extends JComponent implements SwingConstants, Accessible

**Java JLabel Example**

import javax.swing.\*;

class LabelExample

{

public static void main(String args[])

{

JFrame f= new JFrame("Label Example");

JLabel l1,l2;

l1=new JLabel("First Label.");

l1.setBounds(50,50, 100,30);

l2=new JLabel("Second Label.");

l2.setBounds(50,100, 100,30);

f.add(l1); f.add(l2);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

}

=====================================================================================

**Java swing JTextField**

The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.

**//Java JTextField Example**

import javax.swing.\*;

class TextFieldExample

{

public static void main(String args[])

{

JFrame f= new JFrame("TextField Example");

JTextField t1,t2;

t1=new JTextField("Nielit summer training.");

t1.setBounds(50,100, 200,30);

t2=new JTextField("AWT Tutorial");

t2.setBounds(50,150, 200,30);

f.add(t1); f.add(t2);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

}

==================================================================================

**Java swing JTextArea**

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class

import javax.swing.\*;

public class TextAreaExample

{

TextAreaExample(){

JFrame f= new JFrame();

JTextArea area=new JTextArea("Nielit Summer Training");

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 TextAreaExample();

}}

=====================================================================================

**Java JPasswordField Example**

import javax.swing.\*;

public class PasswordFieldExample {

public static void main(String[] args) {

JFrame f=new JFrame("Password Field Example");

JPasswordField value = new JPasswordField();

JLabel l1=new JLabel("Password:");

l1.setBounds(20,100, 80,30);

value.setBounds(100,100,100,30);

f.add(value); f.add(l1);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

}

=================================================================================

**Java JPasswordField Example with ActionListener**

import javax.swing.\*;

import java.awt.event.\*;

public class PasswordFieldExample {

public static void main(String[] args) {

JFrame f=new JFrame("Password Field Example");

final JLabel label = new JLabel();

label.setBounds(20,150, 200,50);

final JPasswordField value = new JPasswordField();

value.setBounds(100,75,100,30);

JLabel l1=new JLabel("Username:");

l1.setBounds(20,20, 80,30);

JLabel l2=new JLabel("Password:");

l2.setBounds(20,75, 80,30);

JButton b = new JButton("Login");

b.setBounds(100,120, 80,30);

final JTextField text = new JTextField();

text.setBounds(100,20, 100,30);

f.add(value);

f.add(l1);

f.add(label);

f.add(l2);

f.add(b);

f.add(text);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String data = "Username " + text.getText();

data += ", Password: "

+ new String(value.getPassword());

label.setText(data);

}

});

}

}

=====================================================================

**Java swing JCheckBox**

The JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a CheckBox changes its state from "on" to "off" or from "off" to "on ".It inherits JToggleButton class.

**Java JCheckBox Example**

import javax.swing.\*;

public class CheckBoxExample

{

CheckBoxExample(){

JFrame f= new JFrame("CheckBox Example");

JCheckBox checkBox1 = new JCheckBox("C++");

checkBox1.setBounds(100,100, 50,50);

JCheckBox checkBox2 = new JCheckBox("Java", true);

checkBox2.setBounds(100,150, 50,50);

f.add(checkBox1);

f.add(checkBox2);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new CheckBoxExample();

}}

==================================================================================

**Java JRadioButton**

The JRadioButton class is used to create a radio button. It is used to choose one option from multiple options. It is widely used in exam systems or quiz.It should be added in ButtonGroup to select one radio button only

**JRadioButton class declaration**

public class JRadioButton extends JToggleButton implements Accessible

**Java JRadioButton Example**

import javax.swing.\*;

public class RadioButtonExample {

JFrame f;

RadioButtonExample(){

f=new JFrame();

JRadioButton r1=new JRadioButton("A) Male");

JRadioButton r2=new JRadioButton("B) Female");

r1.setBounds(75,50,100,30);

r2.setBounds(75,100,100,30);

ButtonGroup bg=new ButtonGroup();

bg.add(r1);bg.add(r2);

f.add(r1);f.add(r2);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String[] args) {

new RadioButtonExample();

}

}

===================================================================================

**Java JRadioButton Example with ActionListener**

import javax.swing.\*;

import java.awt.event.\*;

class RadioButtonExample extends JFrame implements ActionListener{

JRadioButton rb1,rb2;

JButton b;

RadioButtonExample(){

rb1=new JRadioButton("Male");

rb1.setBounds(100,50,100,30);

rb2=new JRadioButton("Female");

rb2.setBounds(100,100,100,30);

ButtonGroup bg=new ButtonGroup();

bg.add(rb1);bg.add(rb2);

b=new JButton("click");

b.setBounds(100,150,80,30);

b.addActionListener(this);

add(rb1);add(rb2);add(b);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

if(rb1.isSelected()){

JOptionPane.showMessageDialog(this,"You are Male.");

}

if(rb2.isSelected()){

JOptionPane.showMessageDialog(this,"You are Female.");

}

}

public static void main(String args[]){

new RadioButtonExample();

}}

================================================================================

**Java swing JComboBox**

The object of Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu. It inherits JComponent class.

**JComboBox class declaration**

public class JComboBox extends JComponent implements ItemSelectable, ListDataListener, ActionListener, Accessible

**Java JComboBox Example**

import javax.swing.\*;

public class ComboBoxExample {

JFrame f;

ComboBoxExample(){

f=new JFrame("ComboBox Example");

String country[]={"India","Aus","U.S.A","England","Newzealand"};

JComboBox cb=new JComboBox(country);

cb.setBounds(50, 50,90,20);

f.add(cb);

f.setLayout(null);

f.setSize(400,500);

f.setVisible(true);

}

public static void main(String[] args) {

new ComboBoxExample();

}

}

===================================================================================

**Java Swing JTable**

The JTable class is used to display data in tabular form. It is composed of rows and columns.

**Java JTable Example**

import javax.swing.\*;

public class TableExample {

JFrame f;

TableExample(){

f=new JFrame();

String data[][]={ {"256","Alok","9256"},

{"257","Prashant","3028"},

{"258","Amandeep","6970"}};

String column[]={"ID","NAME","SALARY"};

JTable jt=new JTable(data,column);

jt.setBounds(30,40,200,300);

JScrollPane sp=new JScrollPane(jt);

f.add(sp);

f.setSize(300,400);

f.setVisible(true);

}

public static void main(String[] args) {

new TableExample();

}

}

=====================================================================================

**Java JList**

The object of JList class represents a list of text items. The list of text items can be set up so that the user can choose either one item or multiple items. It inherits JComponent class.

**JList class declaration**

public class JList extends JComponent implements Scrollable, Accessible

**Java JList Example**

import javax.swing.\*;

public class ListExample

{

ListExample(){

JFrame f= new JFrame();

DefaultListModel<String> l1 = new DefaultListModel<>();

l1.addElement("Item1");

l1.addElement("Item2");

l1.addElement("Item3");

l1.addElement("Item4");

JList<String> list = new JList<>(l1);

list.setBounds(100,100, 75,75);

f.add(list);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new ListExample();

}} ……………..

=====================================================================================

**Java JOptionPane**

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The JOptionPane class inherits JComponent class.

**JOptionPane class declaration**

public class JOptionPane extends JComponent implements Accessible

**Java JOptionPane Example: showMessageDialog()**

import javax.swing.\*;

public class OptionPaneExample {

JFrame f;

OptionPaneExample(){

f=new JFrame();

JOptionPane.showMessageDialog(f,"Hello, You have got a surprise.");

}

public static void main(String[] args) {

new OptionPaneExample();

}

}

=====================================================================================

**Java JOptionPane Example: showInputDialog()**

import javax.swing.\*;

public class OptionPaneExample1 {

JFrame f;

OptionPaneExample1(){

f=new JFrame();

String name=JOptionPane.showInputDialog(f,"Enter Name");

}

public static void main(String[] args) {

new OptionPaneExample1();

}

}

===================================================================================

**Java JOptionPane Example: showConfirmDialog()**

import javax.swing.\*;

import java.awt.event.\*;

public class OptionPaneExample2 extends WindowAdapter{

JFrame f;

OptionPaneExample2(){

f=new JFrame();

f.addWindowListener(this);

f.setSize(300, 300);

f.setLayout(null);

f.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

f.setVisible(true);

}

public void windowClosing(WindowEvent e) {

int a=JOptionPane.showConfirmDialog(f,"Are you sure?");

if(a==JOptionPane.YES\_OPTION){

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

}

}

public static void main(String[] args) {

new OptionPaneExample2();

}

}

=====================================================================================

**Java JScrollBar**

The object of JScrollbar class is used to add horizontal and vertical scrollbar. It is an implementation of a scrollbar. It inherits JComponent class.

**JScrollBar class declaration**

public class JScrollBar extends JComponent implements Adjustable, Accessible

Java JScrollBar Example

import javax.swing.\*;

class ScrollBarExample

{

ScrollBarExample(){

JFrame f= new JFrame("Scrollbar Example");

JScrollBar s=new JScrollBar();

s.setBounds(100,100, 50,100);

f.add(s);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new ScrollBarExample();

}}

=====================================================================================

**Java JMenuBar, JMenu and JMenuItem**

The JMenuBar class is used to display menubar on the window or frame. It may have several menus.The object of JMenu class is a pull down menu component which is displayed from the menu bar. It inherits the JMenuItem class.The object of JMenuItem class adds a simple labeled menu item. The items used in a menu must belong to the JMenuItem or any of its subclass.

**Java JMenuItem and JMenu Example**

import javax.swing.\*;

class MenuExample

{

JMenu menu, submenu;

JMenuItem i1, i2, i3, i4, i5;

MenuExample(){

JFrame f= new JFrame("Menu and MenuItem Example");

JMenuBar mb=new JMenuBar();

menu=new JMenu("Menu");

submenu=new JMenu("Sub Menu");

i1=new JMenuItem("Item 1");

i2=new JMenuItem("Item 2");

i3=new JMenuItem("Item 3");

i4=new JMenuItem("Item 4");

i5=new JMenuItem("Item 5");

menu.add(i1); menu.add(i2); menu.add(i3);

submenu.add(i4); submenu.add(i5);

menu.add(submenu);

mb.add(menu);

f.setJMenuBar(mb);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new MenuExample();

}}

=====================================================================================

**Java JPopupMenu**

PopupMenu can be dynamically popped up at specific position within a component. It inherits the JComponent class.

**Java JPopupMenu Example**

import javax.swing.\*;

import java.awt.event.\*;

class PopupMenuExample

{

PopupMenuExample(){

final JFrame f= new JFrame("PopupMenu Example");

final JPopupMenu popupmenu = new JPopupMenu("Edit");

JMenuItem cut = new JMenuItem("Cut");

JMenuItem copy = new JMenuItem("Copy");

JMenuItem paste = new JMenuItem("Paste");

popupmenu.add(cut); popupmenu.add(copy); popupmenu.add(paste);

f.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

popupmenu.show(f , e.getX(), e.getY());

}

});

f.add(popupmenu);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new PopupMenuExample();

}}

=====================================================================================

**Java JSeparator**

The object of JSeparator class is used to provide a general purpose component for implementing divider lines. It is used to draw a line to separate widgets in a Layout. It inherits JComponent class.

**Java JSeparator Example**

import javax.swing.\*;

class SeparatorExample

{

JMenu menu, submenu;

JMenuItem i1, i2, i3, i4, i5;

SeparatorExample() {

JFrame f= new JFrame("Separator Example");

JMenuBar mb=new JMenuBar();

menu=new JMenu("Menu");

i1=new JMenuItem("Item 1");

i2=new JMenuItem("Item 2");

menu.add(i1);

menu.addSeparator();

menu.add(i2);

mb.add(menu);

f.setJMenuBar(mb);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new SeparatorExample();

}}

=====================================================================================

**Java JColorChooser**

The JColorChooser class is used to create a color chooser dialog box so that user can select any color. It inherits JComponent class.

**Java JColorChooser Example**

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

public class ColorChooserExample extends JFrame implements ActionListener {

JButton b;

Container c;

ColorChooserExample(){

c=getContentPane();

c.setLayout(new FlowLayout());

b=new JButton("color");

b.addActionListener(this);

c.add(b);

}

public void actionPerformed(ActionEvent e) {

Color initialcolor=Color.RED;

Color color=JColorChooser.showDialog(this,"Select a color",initialcolor);

c.setBackground(color);

}

public static void main(String[] args) {

ColorChooserExample ch=new ColorChooserExample();

ch.setSize(400,400);

ch.setVisible(true);

ch.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

=====================================================================================

**Java JTabbedPane**

The JTabbedPane class is used to switch between a group of components by clicking on a tab with a given title or icon. It inherits JComponent class.

**Java JTabbedPane Example**

import javax.swing.\*;

public class TabbedPaneExample {

JFrame f;

TabbedPaneExample(){

f=new JFrame();

JTextArea ta=new JTextArea(200,200);

JPanel p1=new JPanel();

p1.add(ta);

JPanel p2=new JPanel();

JPanel p3=new JPanel();

JTabbedPane tp=new JTabbedPane();

tp.setBounds(50,50,200,200);

tp.add("file",p1);

tp.add("computer",p2);

tp.add("manage",p3);

f.add(tp);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String[] args) {

new TabbedPaneExample();

}}

=====================================================================================

**Java JSlider**

The Java JSlider class is used to create the slider. By using JSlider, a user can select a value from a specific range.

**Java JSlider Example**

import javax.swing.\*;

public class Slider1 extends JFrame{

public Slider1 () {

JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 50, 25);

JPanel panel=new JPanel();

panel.add(slider);

add(panel);

}

public static void main(String s[]) {

Slider1 frame=new Slider1();

frame.pack();

frame.setVisible(true);

}

}

=================================================================================================

**Java JSpinner**

The object of JSpinner class is a single line input field that allows the user to select a number or an object value from an ordered sequence.

import javax.swing.\*;

public class SpinnerExample {

public static void main(String[] args) {

JFrame f=new JFrame("Spinner Example");

SpinnerModel value =

new SpinnerNumberModel(5, //initial value

0, //minimum value

10, //maximum value

1); //step

JSpinner spinner = new JSpinner(value);

spinner.setBounds(100,100,50,30);

f.add(spinner);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

}

=================================================================================================

**Java JDialog**

The JDialog control represents a top level window with a border and a title used to take some form of input from the user. It inherits the Dialog class.

Unlike JFrame, it doesn't have maximize and minimize buttons.

**Java JDialog Example**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class DialogExample {

private static JDialog d;

DialogExample() {

JFrame f= new JFrame();

d = new JDialog(f , "Dialog Example", true);

d.setLayout( new FlowLayout() );

JButton b = new JButton ("OK");

b.addActionListener ( new ActionListener()

{

public void actionPerformed( ActionEvent e )

{

DialogExample.d.setVisible(false);

}

});

d.add( new JLabel ("Click button to continue."));

d.add(b);

d.setSize(300,300);

d.setVisible(true);

}

public static void main(String args[])

{

new DialogExample();

}

}

=====================================================================================

**Java JPanel**

The JPanel is a simplest container class. It provides space in which an application can attach any other component. It inherits the JComponents class.It doesn't have title bar.

**Java JPanel Example**

import java.awt.\*;

import javax.swing.\*;

public class PanelExample {

PanelExample()

{

JFrame f= new JFrame("Panel Example");

JPanel panel=new JPanel();

panel.setBounds(40,80,200,200);

panel.setBackground(Color.gray);

JButton b1=new JButton("Button 1");

b1.setBounds(50,100,80,30);

b1.setBackground(Color.yellow);

JButton b2=new JButton("Button 2");

b2.setBounds(100,100,80,30);

b2.setBackground(Color.green);

panel.add(b1); panel.add(b2);

f.add(panel);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

new PanelExample();

}

}

=================================================================================================

**Java JFileChooser**

The object of JFileChooser class represents a dialog window from which the user can select file. It inherits JComponent class.

**Java JFileChooser Example**

import javax.swing.\*;

import java.awt.event.\*;

import java.io.\*;

public class FileChooserExample extends JFrame implements ActionListener{

JMenuBar mb;

JMenu file;

JMenuItem open;

JTextArea ta;

FileChooserExample(){

open=new JMenuItem("Open File");

open.addActionListener(this);

file=new JMenu("File");

file.add(open);

mb=new JMenuBar();

mb.setBounds(0,0,800,20);

mb.add(file);

ta=new JTextArea(800,800);

ta.setBounds(0,20,800,800);

add(mb);

add(ta);

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==open){

JFileChooser fc=new JFileChooser();

int i=fc.showOpenDialog(this);

if(i==JFileChooser.APPROVE\_OPTION){

File f=fc.getSelectedFile();

String filepath=f.getPath();

try{

BufferedReader br=new BufferedReader(new FileReader(filepath));

String s1="",s2="";

while((s1=br.readLine())!=null){

s2+=s1+"\n";

}

ta.setText(s2);

br.close();

}catch (Exception ex) {ex.printStackTrace(); }

}

}

}

public static void main(String[] args) {

FileChooserExample om=new FileChooserExample();

om.setSize(500,500);

om.setLayout(null);

om.setVisible(true);

om.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

=================================================================================================

**Java JLayeredPane**

The JLayeredPane class is used to add depth to swing container. It is used to provide a third dimension for positioning component and divide the depth-range into several different layers.

**Java JLayeredPane Example**

import javax.swing.\*;

import java.awt.\*;

public class LayeredPaneExample extends JFrame {

public LayeredPaneExample() {

super("LayeredPane Example");

setSize(200, 200);

JLayeredPane pane = getLayeredPane();

//creating buttons

JButton top = new JButton();

top.setBackground(Color.white);

top.setBounds(20, 20, 50, 50);

JButton middle = new JButton();

middle.setBackground(Color.red);

middle.setBounds(40, 40, 50, 50);

JButton bottom = new JButton();

bottom.setBackground(Color.cyan);

bottom.setBounds(60, 60, 50, 50);

//adding buttons on pane

pane.add(bottom, new Integer(1));

pane.add(middle, new Integer(2));

pane.add(top, new Integer(3));

}

public static void main(String[] args) {

LayeredPaneExample panel = new LayeredPaneExample();

panel.setVisible(true);

}

}

=================================================================================================

**ToolTip in Java Swing**

This method setToolTipText() is used to set up a tool tip for the component.

**ToolTip Example**

import javax.swing.\*;

public class ToolTipExample {

public static void main(String[] args) {

JFrame f=new JFrame("Password Field Example");

//Creating PasswordField and label

JPasswordField value = new JPasswordField();

value.setBounds(100,100,100,30);

value.setToolTipText("Enter your Password");

JLabel l1=new JLabel("Password:");

l1.setBounds(20,100, 80,30);

//Adding components to frame

f.add(value); f.add(l1);

f.setSize(300,300);

f.setLayout(null);

f.setVisible(true);

}

}

=====================================================================================

**To change TitleBar icon in Java AWT and Swing**

The setIconImage() method of Frame class is used to change the icon of Frame or Window. It changes the icon which is displayed at the left side of Frame or Window.The Toolkit class is used to get instance of Image class in AWT and Swing.

Toolkit class is the abstract super class of every implementation in the Abstract Window Toolkit(AWT). Subclasses of Toolkit are used to bind various components. It inherits Object class.

**Example to change TitleBar icon in Java AWT**

import java.awt.\*;

class IconExample {

IconExample(){

Frame f=new Frame();

Image icon = Toolkit.getDefaultToolkit().getImage("F:\\nielit.jpeg");

f.setIconImage(icon);

f.setLayout(null);

f.setSize(400,400);

f.setVisible(true);

}

public static void main(String args[]){

new IconExample();

}

}

=====================================================================================**Displaying image in swing:**

For displaying image, we can use the method drawImage() of Graphics class.

**Syntax of drawImage() method:**

public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer): is used to draw the specified image.

**Example to draw image:-**

import java.awt.\*;

import javax.swing.JFrame;

public class MyCanvas extends Canvas{

public void paint(Graphics g) {

Toolkit t=Toolkit.getDefaultToolkit();

Image i=t.getImage("ab.jpeg");

g.drawImage(i, 120,100,this);

}

public static void main(String[] args) {

MyCanvas m=new MyCanvas();

JFrame f=new JFrame();

f.add(m);

f.setSize(400,400);

f.setVisible(true);

}

}

=====================================================================================