Java means DURGA SOFT..

CORE JAVA

Material

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Java.Awt package

- Abstract Window Tool kit is an **API** it supports graphical user interface programming.
- By using java.awt package we are able to develop the components like
  - TextFiled , Label, Button ,Checkbox ,RadioButton…..etc
- AWT components are platform dependent it displays the application according to the view of operating system.
- By using java.awt package we are able to prepare static components to provide the dynamic nature to the component use **java.awt.event** package.**(it is a sub package of java.awt).**
-

1. This application not providing very good look and feel hence the normal users facing problem with these types of applications.
2. By using AWT we are preparing application these applications are called console based or CUI application.

Note

Java.awt package is used to prepare static components.
Java.awt.event package is used to provide the life to the static components.

GUI(graphical user interface):-

1. It is a mediator between end user and the program.
2. AWT is a package it will provide very good predefined support to design GUI applications.

component :-

- ✓ The root class of java.awt package is Component class.
- ✓ Component is an object which is displayed pictorially on the screen.

Ex:- Button,Label,TextField……etc

Container:-

- it is a component in awt that contains another components like Button,TextField…etc
- Container is a sub class of Component class.
- The classes that extends container classes those classes are containers such as Frame, Dialog and Panel.

Event:-

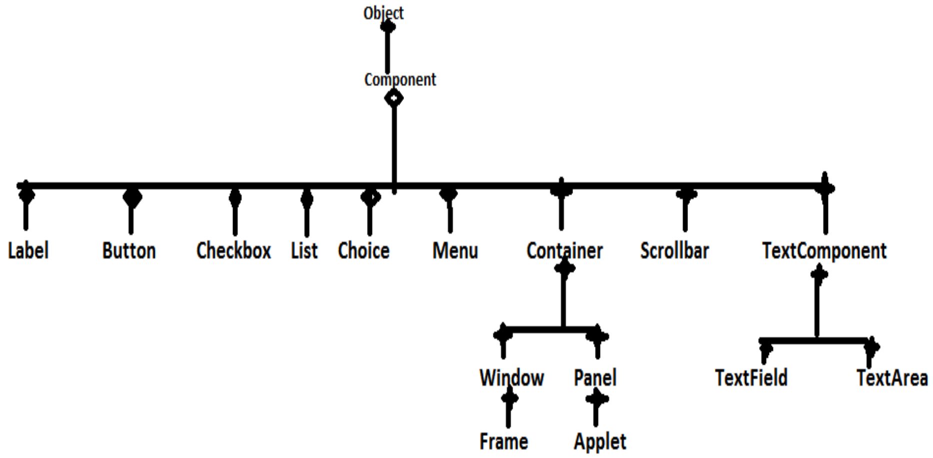
The event nothing but a action generated on the component or the change is made on the state of the object.

Ex:-

Button clicked, Checkboxchecked, Itemselected in the list, Scrollbar scrolled horizontal/vertically.

Classes of AWT:-

The classes present in the AWT package.



Java.awt.Frame:-

Frame is a Basic component in AWT, it contains other components like Button, TextField...etc. There are two approaches to create a frame

1) ***By extending Frame class.***
2) ***By creating Object of Frame class.***

Constructors:-

Frame f=new Frame();
Frame f=new Frame("MyFrame");

Characteristics of the Frame:-

- ✓ When we create a Frame class object the Frame will be created automatically with invisible mode so to provide visible nature to the frame use setVisible() method of Frame class.
  public void setVisible(boolean b)
  where b==true visible mode b==false means invisible mode.
- ✓ When we created a Frame the initial size of the Frame is 0 pixel heights& 0 pixel width so it is not visible to use.To provide particular size to the Frame we have to use setSize() method.
  public void setSize(int width,int height)
- ✓ To provide title to the Frame use.***public void setTitle(String Title)***
- ✓ When we create a Frame, the default background color of the Frame is white. If you want to provide particular color to the Frame we have to use the following method.
  public void setBackground(color c)

Example-1 :- creation of Frame By creating Object of Frame class.

```
import java.awt.*;
class Demo
{	public static void main(String[] args)
	{	Frame f=new Frame();		//frame creation
		f.setVisible(true);		//now frame is visible by default not visible
		f.setSize(400,400);		//set the size of the frame
		f.setBackground(Color.red);	//set the background
		f.setTitle("myframe");		//set the title of the frame
	}
};
```

CRATION OF FRAME BY TAKING USER DEFINED CLASS*

```
import java.awt.*;
class MyFrame extends Frame
{	MyFrame()
	{	setVisible(true);
		setSize(500,500);
		setTitle("myframe");
		setBackground(Color.red);
	}
}
class Demo
{	public static void main(String[] args)
	{	MyFrame f=new MyFrame();
	}
};
```

To display text on the screen:-

1. If you want to display some textual message or some graphical shapes on the Frame then we have to override paint(), which is present in the Frame class.
   public void paint(Graphics g)

2. To set a particular font to the text,we have to use Font class present in java.awt package
   ```
           Font f=new Font(String type,int style,int size);
               Ex: Font f= new Font("arial",Font.Bold,30);
   ```

Ex :-

```
import java.awt.*;
class Test extends Frame
{
        public static void main(String[] args)
        {
                Test t=new Test();
                t.setVisible(true);
                t.setSize(500,500);
                t.setTitle("myframe");
                t.setBackground(Color.red);
        }
        public void paint(Graphics g)
        {
                Font f=new Font("arial",Font.ITALIC,25);
                g.setFont(f);
                g.drawString("hi ratan how r u",100,100);
        }
}
```

Note:-

1. When we create a MyFrame class constructor,jvm executes MyFrame class construcor just before this JVM has to execute Frame class zero argument constructor.
2. In Frame class zero argument constructor repaint() method will be executed, it will access predefined Frame class paint() method. But as per the requirement overriding paint() method will be executed.
3. Therefore the paint() will be executed automatically at the time of Frame creation.

Preparation of the components:-

Label: -

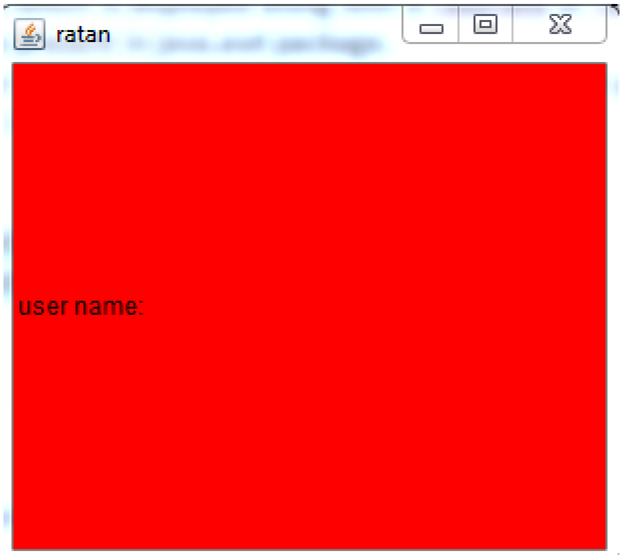
1) Label is a constant text which is displayed along with a TextField or TextArea.
2) Label is a class which is present in java.awt package.
3) To display the label we have to add that label into the frame for that purpose we have to use add() method present in the Frame class.

Constructor:-

```
                Label l=new Label();
        Label l=new Label(“user name”);
```

Ex :-

```
import java.awt.*;
class Test
{
        public static void main(String[] args)
        {
                Frame f=new Frame();
                f.setVisible(true);
                f.setTitle("ratan");
                f.setBackground(Color.red);
                f.setSize(400,500);
        Label l=new Label("user name:");
                f.add(l);
        }
}
```



TextField:-

- ➢ TextField is an editable area and it is possible to provide single line of text.
- ➢ Enter Button doesn’t work on TextField.
    - To set Text to the textarea we have to use ***t.setText(“Sravya”);***
    - To get the text form the TextArea we have to use ***String s=t.getText();***
    - To append text into the TextArea***t.appendText("ratan");***

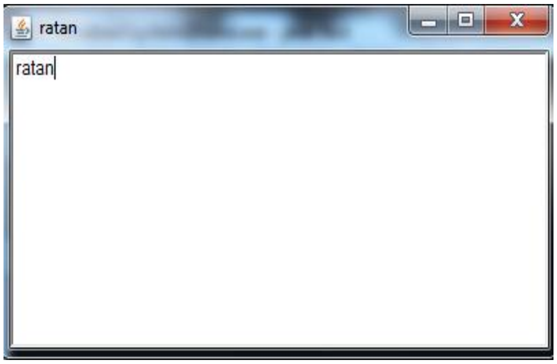
Constructor:-

```
        TextFiled tx=new TextFiled();
        TextField tx=new TextField(“ratan”);
```

Ex :-

```
import java.awt.*;
class Test
{       public static void main(String[] args)
        {       Frame f=new Frame();
                f.setVisible(true);
```

```
                        f.setTitle("ratan");
                        f.setBackground(Color.red);
                        f.setSize(400,500);
//TextField tx=new TextField();  empty TextField
TextField tx=new TextField("ratan");
//TextField with data
                        f.add(tx);
            }
```



ratan

```
}
```

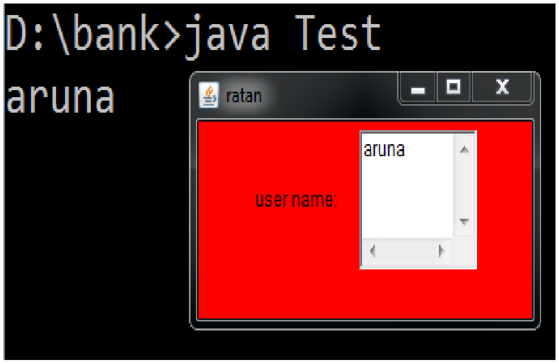
TextArea:-*TextArea is a Editable Area&enter button will work on TextArea.*

TextArea t=new TextArea();

TextArea t=new TextArea(int rows,int columns);

- ✓ To set Text to the textarea we have to use ***ta.setText("Sravya");***
- ✓ To get the text form the TextArea we have to use ***String s=ta.getText();***
- ✓ To append the text into the TextArea*use* **ta.appendText("ratan");**

```
import java.awt.*;
class Test
{           public static void main(String[] args)
            {           Frame f=new Frame();
                        f.setVisible(true);
                        f.setTitle("ratan");
                        f.setBackground(Color.red);
                        f.setSize(400,500);
                        f.setLayout(new FlowLayout());
                        Label l=new Label("user name:");
                        TextArea tx=new TextArea(4,10);//4 character height 10 character width
                        tx.appendText("ratan");
                        tx.setText("aruna");
                        System.out.println(tx.getText());
                        f.add(l);
                        f.add(tx);
                        }
}
```



```
D:\bank>java Test
aruna
```

aruna

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...
JAVA MEANS DURGASOFT
INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE
AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS
#202 2nd FLOOR
www.durgasoft.com
040-64512786
+91 9246212143
+91 8096969696

Choice:-*List is allows to select multiple items but choice is allow to select single Item.*

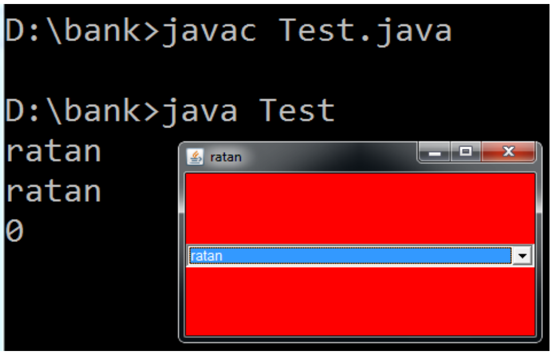
Choice ch=new Choice();

Methods :-

1. To add items to the choice use add() method.
2. To remove item from the choice based on String use remove()method. ***choice.remove("HYD");***
3. To remove the item based on the index position use ***choice.remove(2);***
4. To remove the all elements***ch.removeAll();***
5. To inset the data into the choice based on the particular position.***choice.insert(2,"ratan");***
6. To get selected item from the choice use ***String s=ch.getSelectedItem();***
7. To get the selected item index number use ***int a=ch.getSelectedIndex();***

ex:-

```
import java.awt.*;
class Test
{	public static void main(String[] args)
	{	Frame f=new Frame();
		f.setVisible(true);
		f.setTitle("ratan");
		f.setBackground(Color.red);
		f.setSize(400,500);
		Choice ch=new Choice();
		ch.add("c");
		ch.add("cpp");
		ch.add("java");
		ch.add(".net");
		ch.remove(".net");
		ch.remove(0);
		ch.insert("ratan",0);
		f.add(ch);
System.out.println(ch.getItem(0));
System.out.println(ch.getSelectedItem());
System.out.println(ch.getSelectedIndex());
		//ch.removeAll();
	}
```



```
D:\bank>javac Test.java

D:\bank>java Test
ratan
ratan
0
```

```
}
```

List: List is providinglist of options to select.Based on your requirement we can select any number of elements. To add the List to the frame we have to use add() method.

CONSTRUCTOR:-

List l=new List();It will creates the list by default size is four elements.

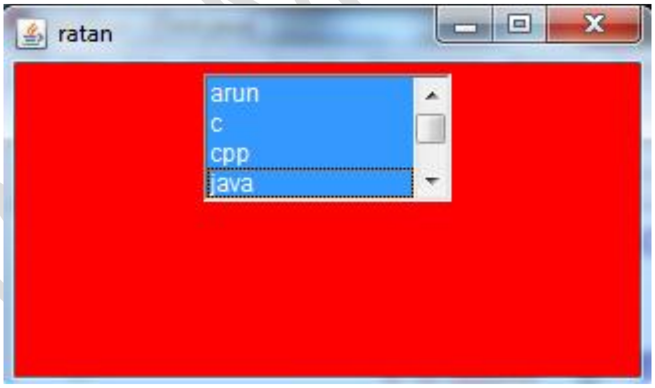
List l=new List(3);It will display the three items size and it is allow selecting the only single item.

List l=new List(5,true);It will display the five items and it is allow selecting the multiple items.

Methods:-

- ✓ To add the elements to the List use ***list.add("c");***
- ✓ To add the elements to the List at specified index ***list.add("ratan",0);***
- ✓ To remove element from the List use ***list.remove("c");***
- ✓ To get selected item from the List use ***String x=l.get SelectedItem();***
- ✓ To get selected items from the List we have to use ***String[] x=s.getSelectedItems()***

```
import java.awt.*;
class Test
{       public static void main(String[] args)
        {       Frame f=new Frame();
                f.setVisible(true);
                f.setTitle("ratan");
                f.setBackground(Color.red);
                f.setSize(400,500);
                f.setLayout(new FlowLayout());
                List l=new List(4,true);
                l.add("c");             l.add("cpp");           l.add("java");          l.add(".net");
                l.add("ratan");         l.add("arun",0);        l.remove(0);            f.add(l);
                System.out.println(l.getSelectedItem());
                }
}
```



Checkbox: - The user can select more than one checkbox at a time.

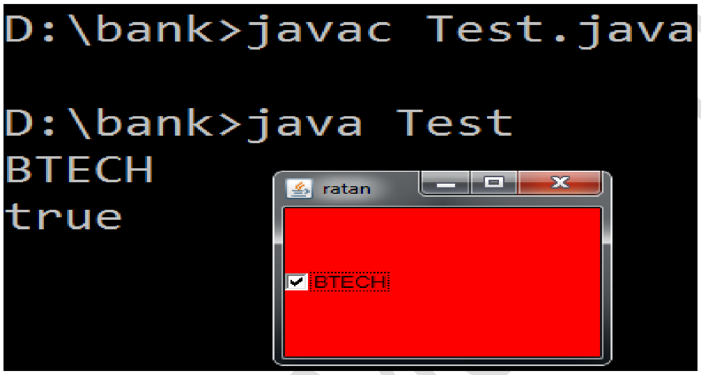
1) Checkbox cb1=new CheckBox();
2) Checkbox cb2=new CheckBox("MCA");
3) Checkbox cb3=new CheckBox("BSC",true);

Methods:-

1. To set a label to the CheckBox explicitly use***cb.setLabel("BSC");***
2. To get the label of the checkbox use ***String str=cb.getLabel();***
3. To get state of the CheckBox use***Boolean b=ch.getState();***

Ex:-

```
import java.awt.*;
class Test
{           public static void main(String[] args)
            {           Frame f=new Frame();
                        f.setVisible(true);
                        f.setTitle("ratan");
                        f.setBackground(Color.red);
                        f.setSize(400,500);
                        Checkbox cb1=new Checkbox("BTECH",true);
                        f.add(cb1);
                        System.out.println(cb1.getLabel());
                        System.out.println(cb1.getState());
                        }
}
```



```
D:\bank>javac Test.java

D:\bank>java Test
BTECH
true
```

RADIO BUTTON:

- ✓ AWT doesnot provide any predefined support to create RadioButtons.
- ✓ It is possible to select Only item from group of items and we are able to create RadioButton by using two classes.
  - ○ CheckBoxgroup
  - ○ CheckBox

step 1:-Create CheckBox group object.   ***CheckBoxGroup cg=new CheckBoxGroup();***

step 2:- pass Checkboxgroup object to the Checkbox class argument.

CheckBox cb1=new CheckBox("male",cg,false);
CheckBox cb2=new CheckBox("female",cg,false);

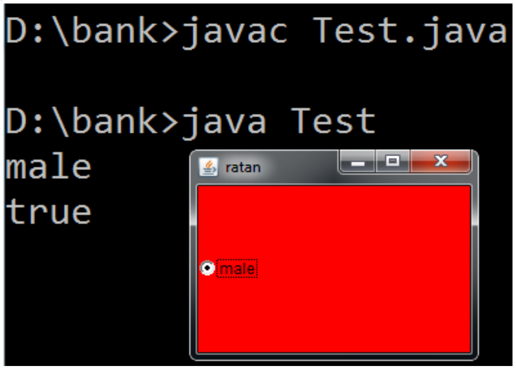
Methods:-

1) To get the status of the RadioButton use ***String str=Cb.getState();***
2) To get Label of the RadioButton use ***String str=getLabel().***

Ex:-

```
import java.awt.*;
class Test
{           public static void main(String[] args)
```

```
          {         Frame f=new Frame();
                    f.setVisible(true);
                    f.setTitle("ratan");
                    f.setBackground(Color.red);
                    f.setSize(400,500);
                    CheckboxGroup cg=new CheckboxGroup();
                    Checkbox cb1=new Checkbox("male",cg,true);
                    f.add(cb1);
                    System.out.println(cb1.getLabel());
                    System.out.println(cb1.getState());
                    }
}
```



```
D:\bank>javac Test.java

D:\bank>java Test
male
true
```

Layout Managers:-

```
import java.awt.*;
class Test
{         public static void main(String[] args)
          {         Frame f=new Frame();
                    f.setVisible(true);
                    f.setTitle("ratan");
                    f.setBackground(Color.red);
                    f.setSize(400,500);
                    Label l1=new Label("user name:");
                    TextField tx1=new TextField();
                    Label l2=new Label("password:");
                    TextField tx2=new TextField();
                    Button b=new Button("login");
                    f.add(l1);              f.add(tx1);              f.add(l2);
                    f.add(tx1);             f.add(b);
          }
}
```

Event delegation model:-

1. When we create a component the components visible on the screen but it is not possible to perform any action for example button.
2. Whenever we create a Frame it can be minimized and maximized and resized but it is not possible to close the Frame even if we click on Frame close Button.
3. The Frame is a static component so it is not possible to perform actions on the Frame.

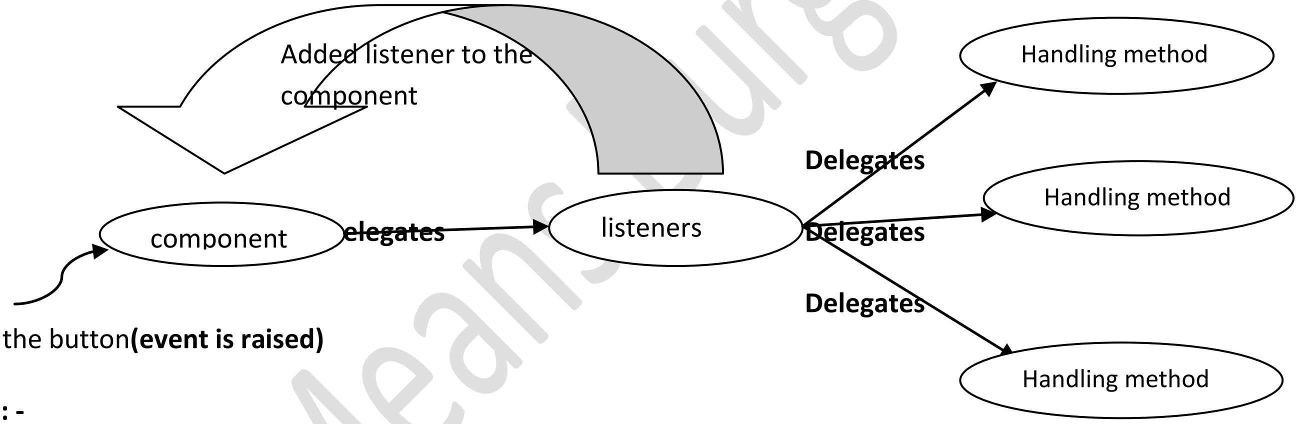
4. To make static component into dynamic component we have to add some actions to the Frame.
5. To attach actions to the Frame component we need event delegation model.

Whenever we click on button no action will be performed clicking like this is called event.

Event: - Event is nothing but a particular action generated on the particular component.

1. When an event generates on the component the component is unable to respond because component can't listen the event.
2. To make the component listen the event we have to add listeners to the component.
3. Wherever we are adding listeners to the component the component is able to respond based on the generated event.
4. A listener is a interface which contain abstract methods and it is present in java.awt.event package
5. The listeners are different from component to component.

A component delegate event to the listener and listener is designates the event to appropriate method by executing that method only the event is handled.This is called Event Delegation Model.



Click the button**(event is raised)**

Note: -

To attach a particular listener to the Frame we have to use following method

Public void AddxxxListener(xxxListener e)

Where xxx may be ActionListener,windowListener

The Appropriate Listener for the Frame is “windowListener”

ScrollBar:-

1. By using ScrollBar we can move the Frame up and down.
   ScrollBar s=new ScrollBar(int type)

Type of scrollbar

1. VERTICAL ScrollBar
2. HORIZONTAL ScrollBar

To create a HORIZONTAL ScrollBar**:-**

ScrollBar sb=new ScrollBar(ScrollBar.HORIZONTAL);
To get the current position of the scrollbar we have to use the following method.
public int getValue()

To create a VERTICAL ScrollBar**:-**

ScrollBar sb=new ScrollBar(ScrollBar.VERTICAL);

Appropriate Listeners for Components:-

GUI Component	Event Name	Listner Name	Lisener Methods
1.Frame	Window Event	Window Listener	1.Public Void WindowOpened(WindowEvent e) 2.Public Void WindowActivated(WindowEvent e) 3.Public Void WindowDeactivated(WindowEvent e) 4.Public Void WindowClosing(WindowEvent e) 5.Public Void WindowClosed(WindowEvent e) 6.Public Void WindowIconfield(WindowEvent e) 7.Public Void WindowDeiconified(WindowEvent e)
2.Textfield	ActionEvent	ActionListener	1.Public Void Actionperformed(ActionEvent ae)
3.TextArea	ActionEvent	ActionListener	1.Public Void Actionperformed(ActionEvent ae)
4.Menu	ActionEvent	ActionListener	1.Public Void Actionperformed(ActionEvent ae)
5.Button	ActionEvent	ActionListener	1.Public Void Actionperformed(ActionEvent ae)
6.Checkbox	ItemEvent	ItemListener	1.Public Void ItemStatechanged(ItemEvent e)
7.Radio	ItemEvent	ItemListener	1.Public Void ItemStatechanged(ItemEvent e)
8.List	ItemEvent	ItemListener	1.Public Void ItemStatechanged(ItemEvent e)
9.Choice	ItemEvent	ItemListener	1.Public Void ItemStatechanged(ItemEvent e)
10.Scrollbar	AdjustmentEvent	AdjustmentListener	1.Public Void AdjustementValueChanged (AdjustementEvent e)
11.Mouse	MouseEvent	MouseListener	1.Public Void MouseEntered(MouseEvent e) 2.Public Void MouseExited(MouseEvent e) 3.Public Void MousePressed(MouseEvent e) 4.Public Void MouseReleased(MouseEvent e) 5.Public Void MouseClicked(MouseEvent e)
12.Keyboard	KeyEvent	KeyListener	1.Public Void KeyTyped(KeyEvent e) 2.Public Void KeyPressed(KeyEvent e) 3.Public Void KeyReleased(KeyEvent e)

PROVIDING CLOSING OPTION TO THE FRAME*

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame
{       MyFrame()
        {       this.setSize(400,500);
                this.setVisible(true);
                this.setTitle("myframe");
                this.addWindowListener(new myclassimpl());
        }
}
class myclassimpl implements WindowListener
{       public void windowActivated(WindowEvent e)
        {       System.out.println("window activated");
        }
        public void windowDeactivated(WindowEvent e)
        {       System.out.println("window deactivated");
        }
        public void windowIconified(WindowEvent e)
        {       System.out.println("window iconified");
        }
        public void windowDeiconified(WindowEvent e)
        {       System.out.println("window deiconified");
        }
        public void windowClosed(WindowEvent e)
        {       System.out.println("window closed");
        }
        public void windowClosing(WindowEvent e)
        {       System.exit(0);
        }
        public void windowOpened(WindowEvent e)
        {       System.out.println("window Opened");
        }
};
class Demo
{       public static void main(String[] args)
        {       MyFrame f=new MyFrame();
        }
};
```

PROVIDING CLOSEING OPTION TO THE FRAME*

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame
{           MyFrame()
            {           this.setVisible(true);
                        this.setSize(500,500);
                        this.setBackground(Color.red);
                        this.setTitle("rattaiah");
                        this.addWindowListener(new Listenerimpl());
            }
};
class Listenerimpl extends WindowAdapter
{           public void windowClosing(WindowEvent we)
            {           System.exit(0);
            }
};
class Demo
{           public static void main(String[] args)
            {           MyFrame f=new MyFrame();
            }
};
```

Note ;- by using WindowAdaptor class we can close the frame. Internally WindowAdaptor class implements WindowListener interface. Hence WindowAdaptor class contains empty implementation of abstract methods.

PROVIDING CLOSEING OPTION THE FRAME***

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame
{           MyFrame()
            {           this.setVisible(true);
                        this.setSize(500,500);
                        this.setBackground(Color.red);
                        this.setTitle("rattaiah");
                        this.addWindowListener(new WindowAdapter()
                        {
            public void windowClosing(WindowEvent we)
            {           System.exit(0);
            }           });
            }
}
class FrameEx
{           public static void main(String[] args)
            {                       MyFrame f=new MyFrame();
            }
};
```

WRITE SOME TEXT INTO THE FRAME*****

```
import java.awt.*;
class MyFrame extends Frame
{       MyFrame()
        {       this.setVisible(true);
                this.setSize(500,500);
                this.setBackground(Color.red);
                this.setTitle("rattaiah");
        }
        public void paint(Graphics g)
        {       Font f=new Font("arial",Font.BOLD,20);
                g.setFont(f);
                this.setForeground(Color.green);
                g.drawString("HI BTECH ",100,100);
                g.drawString("good boys &",200,200);
                g.drawString("good girls",300,300);
        }
}
class FrameEx
{       public static void main(String[] args)
        {       MyFrame f=new MyFrame();
        }
};
```

********LAYOUT MACHANISUMS   FLOWLAYOUT*********

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame
{       Label l1,l2;
        TextField tx1,tx2;
        Button b;
        MyFrame()
        {       this.setVisible(true);
                this.setSize(340,500);
                this.setBackground(Color.green);
                this.setTitle("rattaiah");
                l1=new Label("user name:");             l2=new Label("password:");
                tx1=new TextField(25);          tx2=new TextField(25);
                b=new Button("login");
                tx2.setEchoChar('*');
                this.setLayout(new FlowLayout());
                this.add(l1);     this.add(tx1);    this.add(l2);     this.add(tx2);          this.add(b);
                }
}
class Demo
{       public static void main(String[] args)
        {       MyFrame f=new MyFrame();
        }
};
*****BORDERLAYOUT**********
import java.awt.*;
class MyFrame extends Frame
{       Button b1,b2,b3,b4,b5;
        MyFrame()
        {       this.setBackground(Color.green);
                this.setSize(400,400);
                this.setVisible(true);
                this.setLayout(new BorderLayout());
                 b1=new Button("Boys");
                 b2=new Button("Girls");
                 b3=new Button("management");
                 b4=new Button("Teaching Staff");
                 b5=new Button("non-teaching staff");
                this.add("North",b1);           this.add("Center",b2);
                this.add("South",b3);           this.add("East",b4);
                this.add("West",b5);
        }
}
class Demo
{       public static void main(String[] args)
        {       MyFrame f=new MyFrame();
        }
};
********CardLayout*************
import java.awt.*;
class MyFrame extends Frame
{       MyFrame()
        {       this.setSize(400,400);
                this.setVisible(true);
                this.setLayout(new CardLayout());
                Button b1=new Button("button1");
                Button b2=new Button("button2");
                Button b3=new Button("button3");
                Button b4=new Button("button4");
                Button b5=new Button("button5");
                this.add("First Card",b1);              this.add("Second Card",b2);
                this.add("Thrid Card",b3);              this.add("Fourth Card",b4);
                this.add("Fifth Card",b5);
        }
}
class Demo
{       public static void main(String[] args)
        {               MyFrame f=new MyFrame();
        }
```

```
};
```

```
********GRIDLAYOUT**********
import java.awt.*;
class MyFrame extends Frame
{       MyFrame()
        {       this.setVisible(true);
                this.setSize(500,500);
                this.setTitle("rattaiah");
                this.setBackground(Color.red);
                this.setLayout(new GridLayout(4,4));
                for (int i=0;i<10 ;i++ )
                {       Button b=new Button(""+i);
                        this.add(b);
                }
        }
};
class Demo
{       public static void main(String[] args)
        {       MyFrame f=new MyFrame();
        }
};
*********ACTIONLISTENER**********
import java.awt.*;
import java.awt.event.*;
class myframe extends Frame implements ActionListener
{       TextField tx1,tx2,tx3;
                Label l1,l2,l3;
                Button b1,b2;
                int result;
        myframe()
        {       this.setSize(250,400);
                this.setVisible(true);
                this.setLayout(new FlowLayout());
                 l1=new Label("first value");
                 l2=new Label("second value");
                 l3=new Label("result");

                 tx1=new TextField(25);
                tx2=new TextField(25);
                 tx3=new TextField(25);

                b1=new Button("add");
                b2=new Button("mul");

                b1.addActionListener(this);
                b2.addActionListener(this);
                this.add(l1);           this.add(tx1);          this.add(l2);
                this.add(tx2);          this.add(l3);           this.add(tx3);
                this.add(b1);           this.add(b2);
        }
        public void actionPerformed(ActionEvent e)
        {       try{
                int fval=Integer.parseInt(tx1.getText());
                int sval=Integer.parseInt(tx2.getText());
                String label=e.getActionCommand();
                if (label.equals("add"))
                {       result=fval+sval;
                }
                if (label.equals("mul"))
                {       result=fval*sval;
                }
                tx3.setText(""+result);
                }
        catch(Exception ee)
                {
                ee.printStackTrace();
                }
        }
};
class Demo
{       public static void main(String[] args)
        {       myframe f=new myframe();
        }
};
```

*****   LOGIN STATUS*********

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame implements ActionListener
{           Label l1,l2;
            TextField tx1,tx2;
            Button b;
            String status="";
            MyFrame()
            {           setVisible(true);
                        setSize(400,400);
                        setTitle("girls");
                        setBackground(Color.red);
                        l1=new Label("user name:");
                        l2=new Label("password:");
                        tx1=new TextField(25);
                        tx2=new TextField(25);

                        b=new Button("login");
                        b.addActionListener(this);
                        tx2.setEchoChar('*');

                        this.setLayout(new FlowLayout());

                        this.add(l1);
                        this.add(tx1);
                        this.add(l2);
                        this.add(tx2);
                        this.add(b);
            }
            public void actionPerformed(ActionEvent ae)
            {           String    uname=tx1.getText();
                        String upwd=tx2.getText();
                        if (uname.equals("Sravya")&&upwd.equals("dss"))
                        {           status="login success";                }
                        else
                        {           status="login failure";                }
            repaint();
            }
            public void paint(Graphics g)
                        {
                        Font f=new Font("arial",Font.BOLD,30);
                        g.setFont(f);
                        this.setForeground(Color.green);
                        g.drawString("Status:----"+status,50,300);

                        }
}
```

```
class Demo
{		public static void main(String[] args)
		{		MyFrame f=new MyFrame();
		}
};
******MENUITEMS************

import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame implements ActionListener
{
		String label="";
		MenuBar mb;
		Menu m1,m2,m3;
		MenuItem mi1,mi2,mi3;
		MyFrame()
		{		this.setSize(300,300);
				this.setVisible(true);
				this.setTitle("myFrame");
				this.setBackground(Color.green);

				 mb=new MenuBar();
				 this.setMenuBar(mb);

				 m1=new Menu("new");
				 m2=new Menu("option");
				 m3=new Menu("edit");
				mb.add(m1);
				mb.add(m2);
				mb.add(m3);

				mi1=new MenuItem("open");
				mi2=new MenuItem("save");
				mi3=new MenuItem("saveas");

				mi1.addActionListener(this);
				mi2.addActionListener(this);
				mi3.addActionListener(this);

				m1.add(mi1);
				m1.add(mi2);
				m1.add(mi3);
		}
		public void actionPerformed(ActionEvent ae)
		{
				label=ae.getActionCommand();
				repaint();
    }


    public void paint(Graphics g)
    {
        Font f=new Font("arial",Font.BOLD,25);
        g.setFont(f);
        g.drawString("Selected item....."+label,50,200);
    }
}
class Demo
{
    public static void main(String[] args)
    {
        MyFrame f=new MyFrame();
    }
};
```

*****MOUSELISTENER INTERFACE**********

```
import java.awt.*;
import java.awt.event.*;
class myframe extends Frame implements MouseListener
{   String[] msg=new String[5];
    myframe()
    {   this.setSize(500,500);
        this.setVisible(true);
        this.addMouseListener(this);
    }
    public void mouseClicked(MouseEvent e)
    {   msg[0]="mouse clicked......("+e.getX()+","+e.getY()+")";
        repaint();
    }
    public void mousePressed(MouseEvent e)
    {   msg[1]="mouse pressed......("+e.getX()+","+e.getY()+")";
        repaint();
    }
    public void mouseReleased(MouseEvent e)
    {   msg[2]="mouse released......("+e.getX()+","+e.getY()+")";
        repaint();
    }
    public void mouseEntered(MouseEvent e)
    {   msg[3]="mouse entered......("+e.getX()+","+e.getY()+")";
        repaint();
    }
    public void mouseExited(MouseEvent e)
```

```
        {       msg[4]="mouse exited......("+e.getX()+","+e.getY()+")";
                repaint();
        }
        public void paint(Graphics g)
        {       int X=50;
                int Y=100;
                for(int i=0;i<msg.length;i++)
                {       if (msg[i]!=null)
                        {       g.drawString(msg[i],X,Y);
                                Y=Y+50;
                        }
                }
        }
};
class Demo
{       public static void main(String[] args)
        {       myframe f=new myframe();
        }
};
```

*******ITEMLISTENER INTERFACE**********

```
import java.awt.*;
import java.awt.event.*;
class myframe extends Frame implements ItemListener
{       String qual="",gen="";
        Label l1,l2;
        CheckboxGroup cg;
        Checkbox c1,c2,c3,c4,c5;
        Font f;
        myframe()
        {       this.setSize(300,400);
                this.setVisible(true);
                this.setLayout(new FlowLayout());

                 l1=new Label("Qualification:  ");
                 l2=new Label("Gender:  ");

                 c1=new Checkbox("BSC");
                 c2=new Checkbox("BTECH");
                 c3=new Checkbox("MCA");

                 cg=new CheckboxGroup();
                 c4=new Checkbox("Male",cg,false);
                 c5=new Checkbox("Female",cg,true);

                c1.addItemListener(this);
                c2.addItemListener(this);
                c3.addItemListener(this);
                c4.addItemListener(this);
                c5.addItemListener(this);

                this.add(l1);           this.add(c1);           this.add(c2);
                this.add(c3);           this.add(l2);           this.add(c4);
                this.add(c5);
        }
                public void itemStateChanged(ItemEvent ie)
                {       if(c1.getState()==true)
                        {       qual=qual+c1.getLabel()+",";
                        }
                        if(c2.getState()==true)
                        {       qual=qual+c2.getLabel()+",";
                        }
                        if(c3.getState()==true)
                        {       qual=qual+c3.getLabel()+",";
                        }
                        if(c4.getState()==true)
                        {       gen=c4.getLabel();
                        }
                        if(c5.getState()==true)
                        {       gen=c5.getLabel();
                        }
                        repaint();
                }
                public void paint(Graphics g)
                {       Font f=new Font("arial",Font.BOLD,20);
                        g.setFont(f);
                        this.setForeground(Color.green);
                        g.drawString("qualification------>"+qual,50,100);
                        g.drawString("gender-------------->"+gen,50,150);
```

```
            qual="";
            gen="";
        }
}

class rc
{   public static void main(String[] args)
    {   myframe f=new myframe();
    }
};
*********KEYLISTENER INTERFACE***********
import java.awt.*;
import java.awt.event.*;
class myframe extends Frame
{   myframe()
    {   this.setSize(400,400);
        this.setVisible(true);
        this.setBackground(Color.green);
        this.addKeyListener(new keyboardimpl());
    }
};
class keyboardimpl implements KeyListener
{   public void keyTyped(KeyEvent e)
    {   System.out.println("key typed "+e.getKeyChar());
    }
    public void keyPressed(KeyEvent e)
    {   System.out.println("key pressed "+e.getKeyChar());
    }
    public void keyReleased(KeyEvent e)
    {   System.out.println("key released "+e.getKeyChar());
    }
}
class Demo
{   public static void main(String[] args)
    {   myframe f=new myframe();
    }
};
**********CHECK LIST AND CHOICE************
import java.awt.*;
import java.awt.event.*;
class myframe extends Frame implements ItemListener
{   Label l1,l2;
    List l;
    Choice ch;
    String[] tech;
    String city="";
    myframe()
```

```
        {       this.setSize(300,400);
                this.setVisible(true);
                this.setLayout(new FlowLayout());

                 l1=new Label("Technologies:  ");
                 l2=new Label("City:  ");

                 l=new List(3,true);
                 l.add("c");                    l.add("c++");                l.add("java");
                l.addItemListener(this);

                ch=new Choice();
                ch.add("hyd");              ch.add("chenni");                     ch.add("Banglore");
                ch.addItemListener(this);

                this.add(l1);                this.add(l);                 this.add(l2);                this.add(ch);
        }
        public void itemStateChanged(ItemEvent ie)
        {       tech=l.getSelectedItems();
                city=ch.getSelectedItem();
                repaint();
        }
        public void paint(Graphics g)
        {       Font f=new Font("arial",Font.BOLD,20);
                g.setFont(f);
                String utech="";
                for(int i=0;i<tech.length ;i++ )
                {       utech=utech+tech[i]+"   ";                 }
                g.drawString("tech:-------"+utech,50,200);
                g.drawString("city---------"+city,50,300);
                utech="";
        }
}
class Demo
{       public static void main(String[] args)
        {       myframe f=new myframe();
        }
};
*********AdjustmentListener**********
import java.awt.*;
import java.awt.event.*;
class myframe extends Frame implements AdjustmentListener
{       Scrollbar sb;
        int position;
        myframe()
        {       this.setSize(400,400);
                this.setVisible(true);
```

```
                    this.setLayout(new BorderLayout());

                    sb=new Scrollbar(Scrollbar.VERTICAL);
                    this.add("East",sb);

                    sb.addAdjustmentListener(this);
          }
          public void adjustmentValueChanged(AdjustmentEvent e)
          {         position=sb.getValue();
          }
          public void paint(Graphics g)
          {         g.drawString("position:"+position,100,200);
                    repaint();
          }
}
class scrollbarex
{         public static void main(String[] args)
          {         myframe f=new myframe();
          }
};
```

LEARN FROM EXPERTS ...

COMPLETE JAVA

CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...

COMPLETE .NET

C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF

TESTING TOOLS

MANUAL + SELENIUM

ORACLE D2K

MSBI SHARE POINT

HADOOP ANDROID

C, C++, DS, UNIX

CRT & APTITUDE TRAINING

AN ISO 9001:2008 CERTIFIED

DURGA

Software Solutions®

202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,
9246212143, 8096969696

www.durgasoft.com