# Abstract Window Toolkit (AWT)

- AWT (Abstract Window Toolkit)

Java AWT is an API to develop GUI or windows based application using java awt packages. Fig. shows the important classes of java.awt packages.

a] Java AWT componant are platform dependent i.e component are display according to the view of operating system.

b] AWT is a heavy object component i.e it's component uses the resources of system.

The java.awt package provide classes for AWT API such as TextField, label, TextArea, RedioButton, CheckBox, Choice, List etc. Using these components (class) we can create interactive user interface application.

## Classes of AWT (Hirarchy of AWT) :-

Component :-

A component represent an object which displ pictorial representation on the screen.

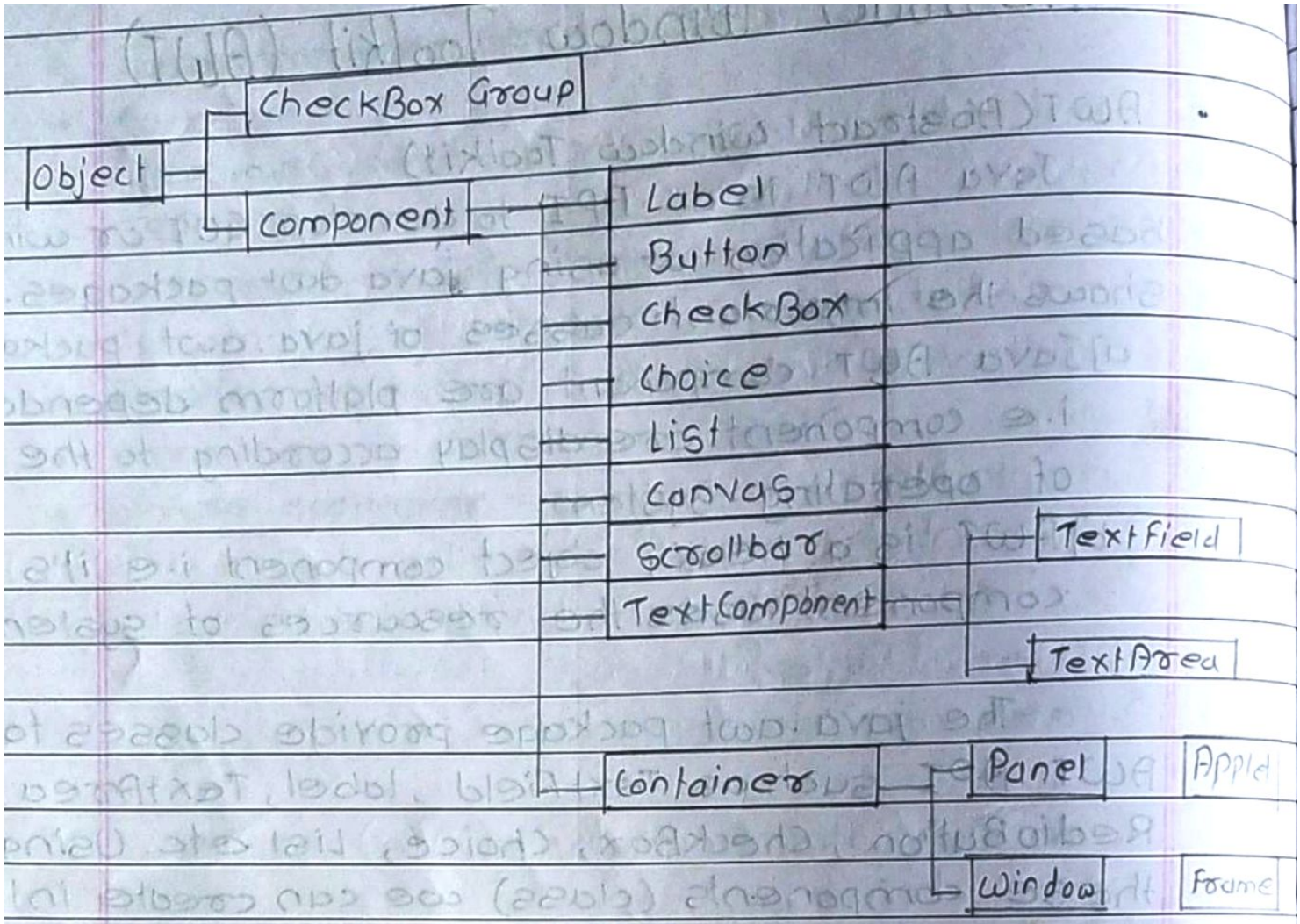
For example :-

Create an object of 'Button' class as

```
Button b=new Button();
```

Here 'b' is the object of 'Button' class in the same way other are the components.

The top of AWT hirarchy the component class i a super class for GUI controls.




* Hierarchy of AWT *

* Container =

Container class is the sub class of component class. It containg another component like buttons, TextField, Labels etc. The following are containers

1] Frame
2] Window
3] Panel

1] Frame = The frame is the container that contain title bar, resize, corner & have manubar. It can have other components like button, TextField etc

2] Window = The window class is basically used to create a top level window. This window directly

set of desktop. The window is the container that has no border & no menubar. You can must use frame for creating windows based application

3) Panel :

The panel is the container that does not contain titlebar and menubar. It can have other components like Button, TextField etc.

Panel is the concreate subclass of container.

* Canvas :

It represents a blank rectangle area on the screen which is used for to draw any application

** Usefull Methods of Component Class :-

| | Method | Description. |
|---|---|---|
| 1] | Public void add (component) | Insert the component on the container. |
| 2] | Public void setsize (int width, int height) | setsize of component. |
| 3] | Public void setlayout (layout manager) | Define layout manager for component. |
| 4] | Public void setvisible (Boolean status) | Change the visibility of components by default false |

** Working with Frames :

Frame is the standard window which is sub-class of window class from AWT hirarchy. Frame becomes a basic component in AWT. The following are the constructor of frame

| Constructor | Description. |
|---|---|
| 1] Frame() | Constructor will create std. Frame that does not contain title |
| 2] Frame(String title) | Constructor will create std. Frame that contain title. |

| Methods | Description |
|---|---|
| 1] Void setsize( int width, int height) | Setting the size of frame height & width |
| 2] Void setTitle(String title) | Setting a frame title |
| 3] Void setvisible : (boolean) | Hiding & Showing the frame by default it is invisible |

example :-

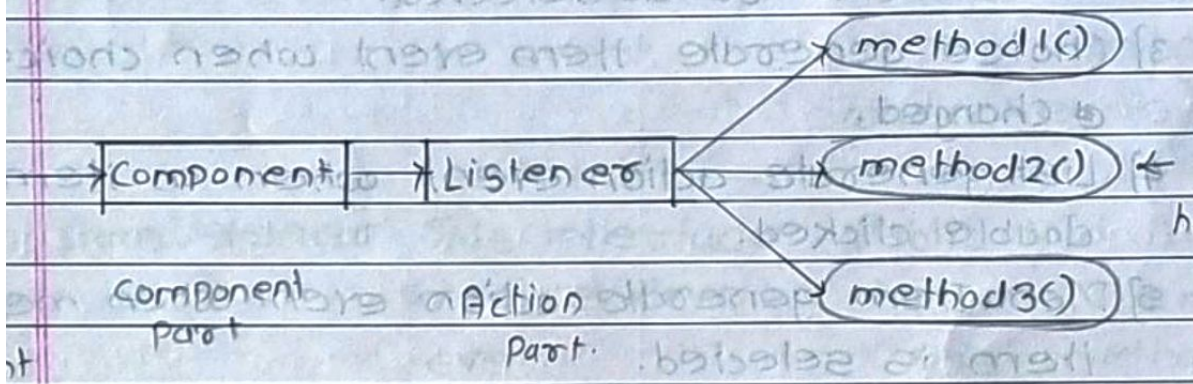
```
import java.awt.*;
Public class frame.demo extends frame
{
    frame f;
    framedemo()
    {
        f=new frame();
        setsize(500, 500);
        setTitle ("Hellow");
        setVisible(true);
    }
Public static void main (String args[])
{
    framedemo f1 = new framedemo();
}
}
```

## Closing Frame :

To window close event, we must implement, windowClosing() method of the windowListener interface inside the windowClosing () we will have to remove the window from the screen.

```
addWindowListener (new WindowAdapter ()
{
    Public void windowClosing (WindowEvent we)
    {
        System.exit(0);
    }
});
```

## Event Delegation Model :-




Event delegation model represents that when an event generated by the user on a component. It is delegated to a listener interface & the listener interface calls a method in represents to the event. Finally, event is handled by the method.

1) Event :

Event is a object that describes what happen on component.