JAVA AWT

* Introduction to AWT :-

- Java AWT (Abstract Window Toolkit) is an API to develop Graphical user interface GUI or Window based applications in Java.
- Java AWT components are platform dependent i.e. components are displayed according to the view of operating System.
- java.awt package is used.
- It is heavy-weight in use, because it contains a large number of classes and methods, which are used for creating and managing GUI.
- java.awt package provides classes for AWT API such as Textfield, Label, TextArea, RadioButton, checkBox, Choice, List etc.

Platform dependent ? An application build on AWT would look like a windows application when it runs on windows, but the same application ~~when~~

would look like a mac, application when runs on mac OS.

* **Advantages:-**

It is easy to use for beginners due to its easy interface.

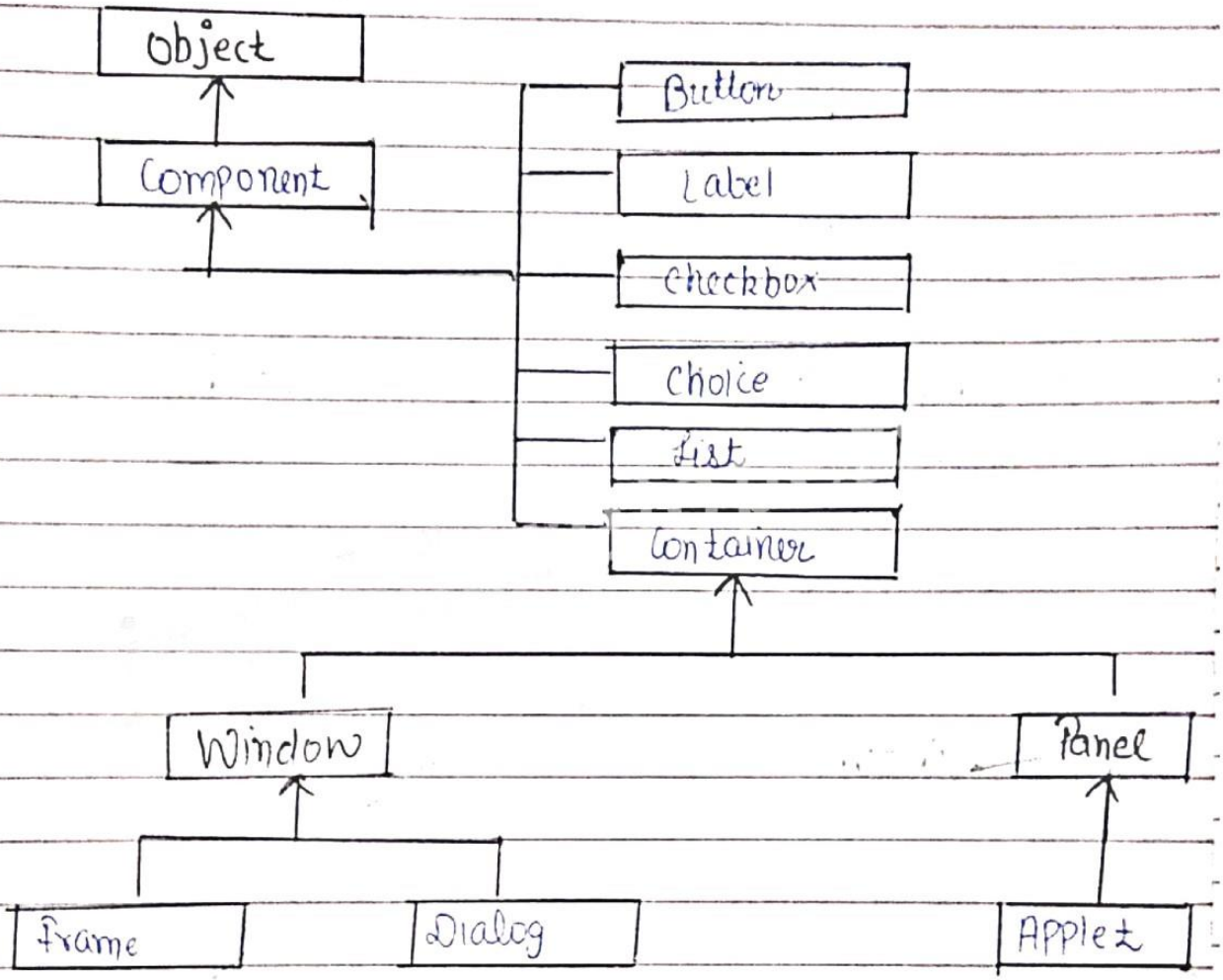
It is dependent on operating system so performance is High.

It takes less memory for development of GUI and executing programs.

* **Disadvantages**

- It is heavyweight in nature.
- It is platform dependent.
- The buttons of AWT does not support pictures.
- AWT Supports limited number of GUI Components.
- It behaves differently in different operating System.

* Java AWT Hierarchy :-



* Components :-

→ All the elements like the button, text fields, scroll bars, etc. are called components.

→ In order to place every component in a particular position on a screen, we need to add them to a container.

add()

__/__/__

- **Container:-**

→ The Container is a component in AWT that can contain another components like buttons, textfields, labels etc

→ The classes that extends Container class are known as container such as Frame, Dialog and Panel.

- There are four types of Container:-
  1. Window
  2. Panel
  3. Frame
  4. Dialog

- **Window:-**

→ The Window is the Container that have no borders and menu bars.

→ you must use frame, dialog for creating a window.

- **Panel:-**

→ The Panel is the Container that doesn't contain title bar and menu bar. Chave Components button, Textfield

- **Frame:-** The frame is the container that contain title bar and can have menu bars.

__/__/____

* Important Methods of Component class:-

Method	Description
public void add (Component c)	insert a component on this component.
public void setsize (int width, int width)	sets the size of component.
public void set Layout ( )	defines the layout manager
public void set Visible (Boolean)	Visibility of the component

Simple Java AWT Example:- (How to Create frame)

(1)

```
import java.awt.*;

public class Myframe
{
  public static void main (String args[])
  {
    Frame f = new Frame ();
    f.setVisible (true);
    f.setSize (500, 300);
    f.setTitle ("jpwebdevelopers");
  }
}
```

__/__/____

(2) By Extending frame Class

```
import java.awt.*;

public class Myframe extends Frame
{
    Myframe()                    // Constructor
    {
        this.setVisible(true);
        this.setSize(500,300);
        this.setTitle("jpwebdevelopes");
    }
    public static void main(String args[])
    {
        Myframe f = new Myframe();
    }
}
```

AWT Example by Inheritance:-

```
import java.awt.*;

public class Example extends Frame
{
    Example()
    {
```

Creating a Button → `Button b = new Button("Click here");`

Setting button Position on screen → `b.setBounds(30,100,80,30);`

adding button into frame → `add(b);`

```
    setSize(300,300);    // frame size 300 width 300 height
    setTitle("JPWEBDEVELOPERS");

    setLayout(null);     // no layout manager
    setVisible(True);    // frame will be visible
}
public static void main(String args[])
{
    Example e = new Example();
}}
```

Output:-