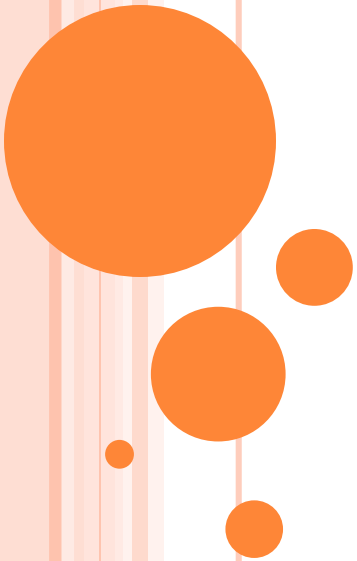


MODULE IV

PART II

INTRODUCTION

- In Java, all programming's can be broadly classified into three categories,
- **the core programming,**
- **applet programming and**
- **AWT programming.**
- Now, AWT programming and applet programming they are alternatively called the **graphics oriented** programming or more precisely we can say window based programming.
- So, here basically we have to develop windows and through windows the programming aspects can be carried out.

Core java

- It's a general term used by Sun Microsystems to describe the standard version of Java (JSE).
- It's the most basic version of Java which sets the foundation for all other editions of Java plus a set of related technologies such as CORBA, Java VM, etc.
- Core Java refers to a collection of libraries rather than just the programming language.
- It's the purest form of Java primarily used for development of general desktop applications.
- Simply speaking, it refers to the subset of Java SE technologies which consists of both general purpose API's and special purpose API's.

Applet

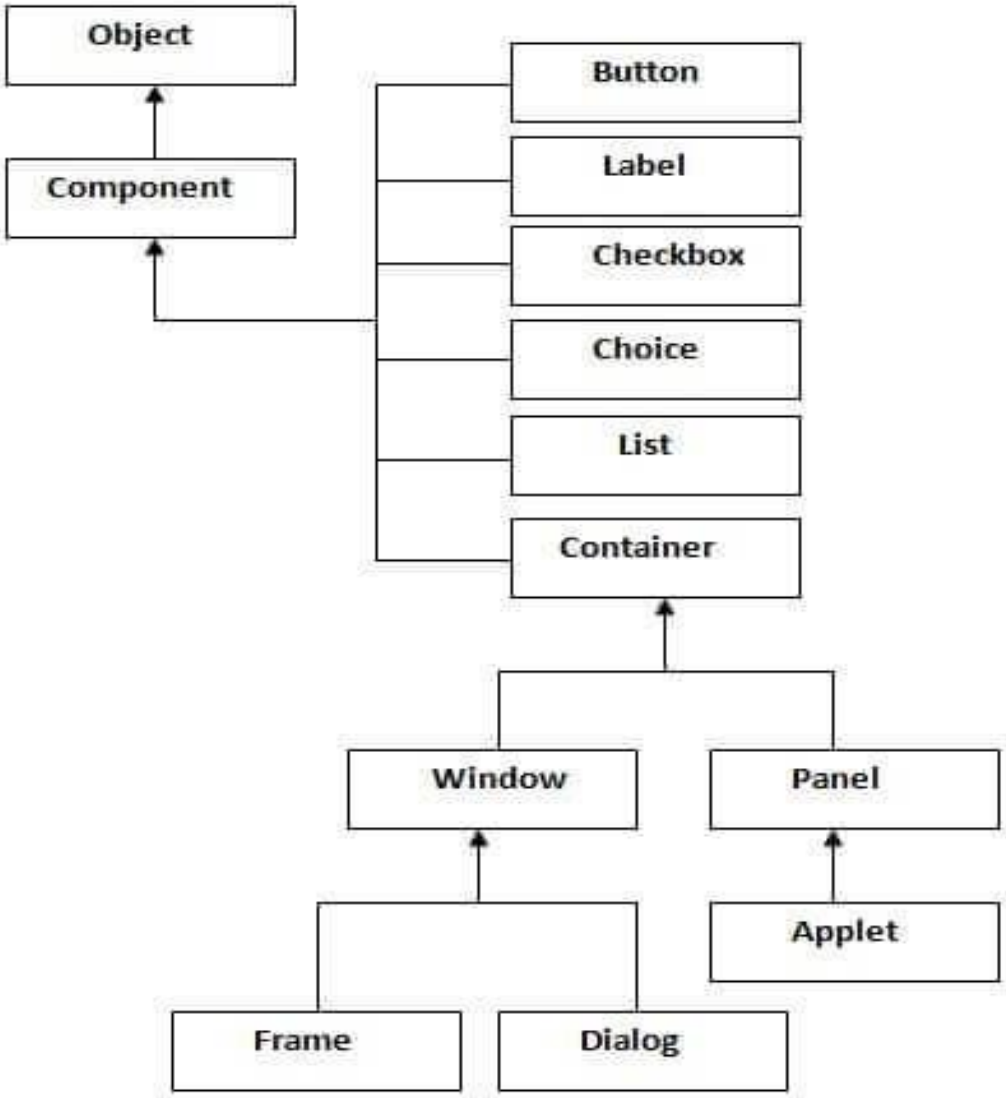
- An **applet** is a Java program that runs in a Web browser.
- An applet can be a fully functional Java application because it has the entire Java API at its disposal.
- There are some important differences between an applet and a standalone Java application, including the following −
  - An applet is a Java class that extends the java.applet.Applet class.
  - A main() method is not invoked on an applet, and an applet class will not define main().
  - Applets are designed to be embedded within an HTML page.
  - When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine.
  - A JVM is required to view an applet. The JVM can be either a plug-in of the Web browser or a separate runtime environment.
  - The JVM on the user's machine creates an instance of the applet class and invokes various methods during the applet's lifetime.
  - Applets have strict security rules that are enforced by the Web browser. The security of an applet is often referred to as sandbox security, comparing the applet to a child playing in a sandbox with various rules that must be followed.
- Applet is not included in your syllabus, but if interested for more reference you can visit the following link:

  https://youtu.be/cC_Ij7WmP_k

JAVA AWT

- **Java AWT** (Abstract Window Toolkit) is *an API to develop 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.**
- AWT is heavyweight i.e. its components are using the resources of OS.
- The java.awt package provides classes for AWT api such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

AWT HIERARCHY



- awt package include many classes for graphics based programming and in each class we have constructors and many useful methods.
- Important classes and methods are included in this pdf.
- For more details visit the following link:
- https://docs.oracle.com/javase/7/docs/api/java/awt/class-use/Component.html

Components and containers

- All the elements like buttons, text fields, scrollbars etc are known as **components**.
- In AWT we have classes for each component as shown in the above diagram.
- To have everything placed on a screen to a particular position, we have to add them to a **container**.
- A **container** is like a screen wherein we are placing components like buttons, text fields, checkbox etc.
- In short a container contains and controls the layout of components.
- A container itself is a component (shown in the above hierarchy diagram) thus we can add a container inside container.

CONTAINERS

- 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.
- **Window**
  - The window is the container that have no borders and menu bars.
  - You must use frame, dialog or another window for creating a window.
- **Panel**
  - The Panel is the container that doesn't contain title bar and menu bars.
  - It can have other components like button, textfield etc.
- **Frame**
  - The Frame is the container that contain title bar and can have menu bars.
  - It can have other components like button, textfield etc.

Useful methods of componenet class

Method	Description
public void add(Component c)	inserts a component on this component.
public void setSize(int width,int height)	sets the size (width and height) of the component.
public void setLayout(LayoutManager m)	defines the layout manager for the component.
public void setVisible(boolean status)	changes the visibility of the component, by default false.

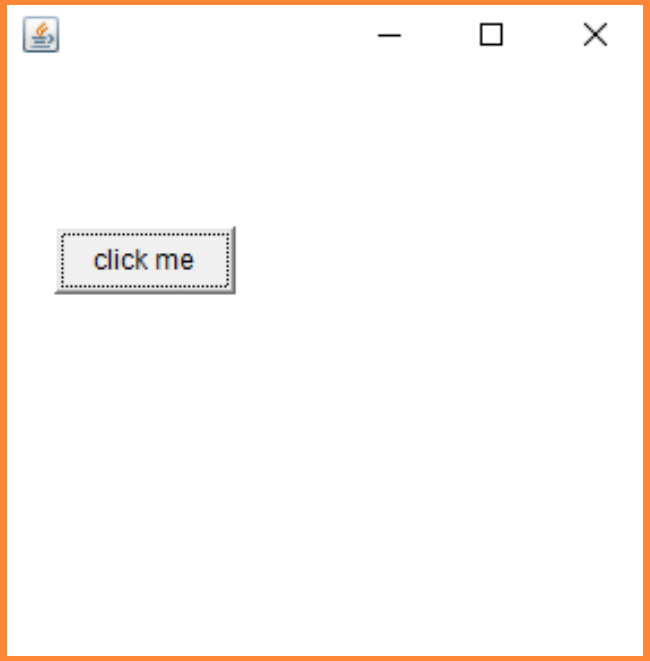
Creating a frame

- There are two ways to create a frame in AWT.
  1. By extending Frame class (inheritance)
  2. By creating the object of Frame class (association)

Example-Frame using inheritance/creating and adding button

```
import java.awt.*;
public class First extends Frame{
First(){
Button b=new Button("click me");  //calling constructor
//of class Button
b.setBounds(40,100,100,40);// setting button position
add(b);//adding button into frame
setSize(400,400);//frame size 300 width and 300 height
setLayout(null);//no layout manager
setVisible(true);//now frame will be visible, by default
not visible
}
public static void main(String args[]){
First f=new First();
}}
```

Output

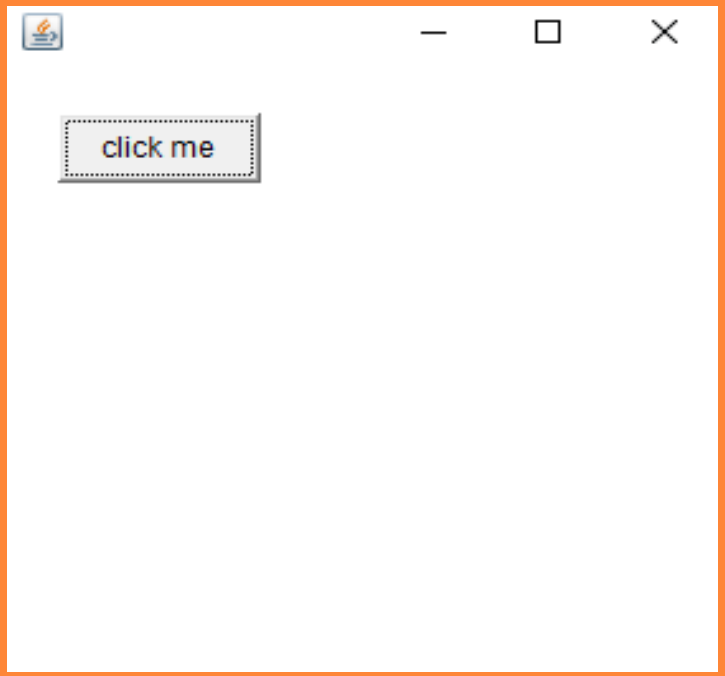


[Back Ground Color filled to highlight the output
Output shown using JDK 13/command propt for execusion/Windows 10]

Example-frame using association and adding a button to the frame

```
import java.awt.*;
public class First{
First(){
Frame f=new Frame();
Button b=new Button("click me");
b.setBounds(30,50,80,30);
f.add(b);
f.setSize(300,300);
f.setLayout(null);
f.setVisible(true);
}
public static void main(String args[]){
First f=new First();
}}
```

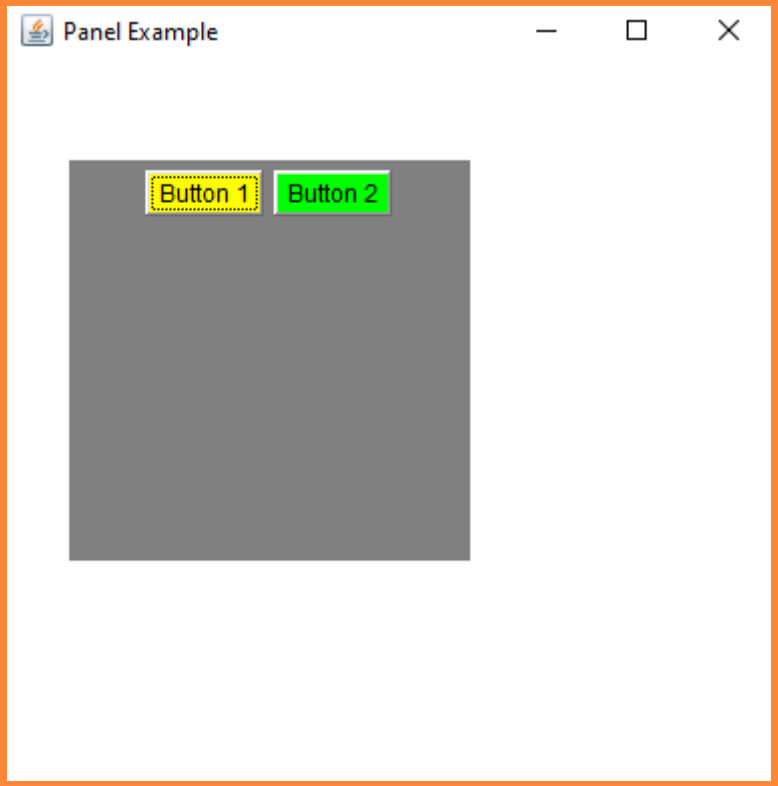
OUTPUT



EXAMPLE –FRAME WITH PANEL

```
import java.awt.*;
public class First {
    First()
      {
      Frame f= new Frame("Panel Example");   //creating frame
      Panel panel=new Panel();  //creating panel
      panel.setBounds(40,80,200,200);
      panel.setBackground(Color.gray);
      Button b1=new Button("Button 1");
      b1.setBounds(50,100,80,30);
      b1.setBackground(Color.yellow);
      Button b2=new Button("Button 2");
      b2.setBounds(100,100,80,30);
      b2.setBackground(Color.green);
      panel.add(b1); panel.add(b2);  //adding buttons to panel
      f.add(panel);  //adding panel to frame
      f.setSize(400,400);
      f.setLayout(null);
      f.setVisible(true);
      }
      public static void main(String args[])
      {
      new First();
      }
}
```

OUTPUT



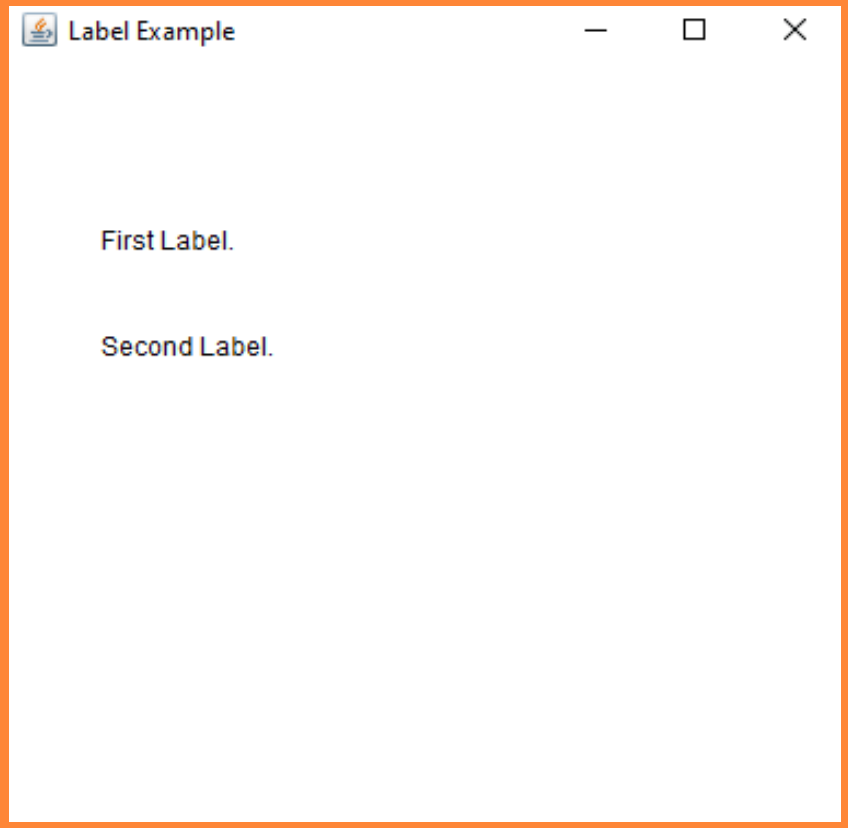
JAVA AWT LABEL

- The object of Label 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.

EXAMPLE-LABEL

```
import java.awt.*;
public class First{
public static void main(String args[]){
    Frame f= new Frame("Label Example");
    Label l1,l2;
    l1=new Label("First Label.");
    l1.setBounds(50,100, 100,30);
    l2=new Label("Second Label.");
    l2.setBounds(50,150, 100,30);
    f.add(l1); f.add(l2);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

OUTPUT



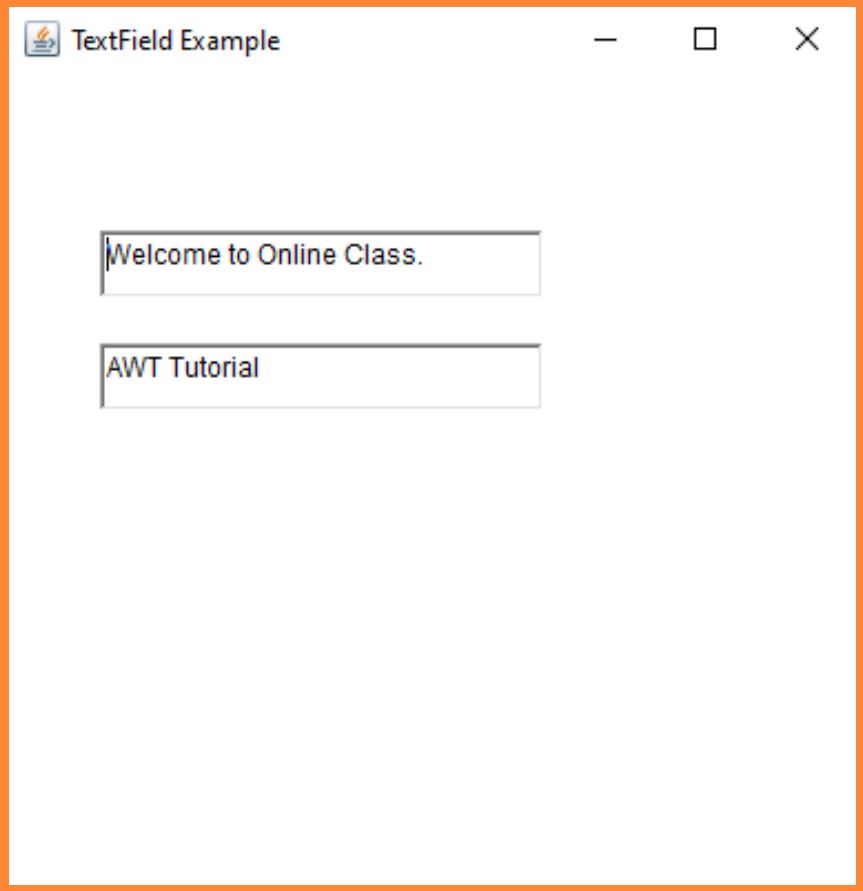
Java AWT TextField

- The object of a TextField class is a text component that allows the editing of a single line text.

EXAMPLE-TEXTFIELD

```
import java.awt.*;
class First{
public static void main(String args[]){
    Frame f= new Frame("TextField Example");
    TextField t1,t2;
    t1=new TextField("Welcome to Online Class.");
    t1.setBounds(50,100, 200,30);
    t2=new TextField("AWT Tutorial");
    t2.setBounds(50,150, 200,30);
    f.add(t1); f.add(t2);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

OUTPUT



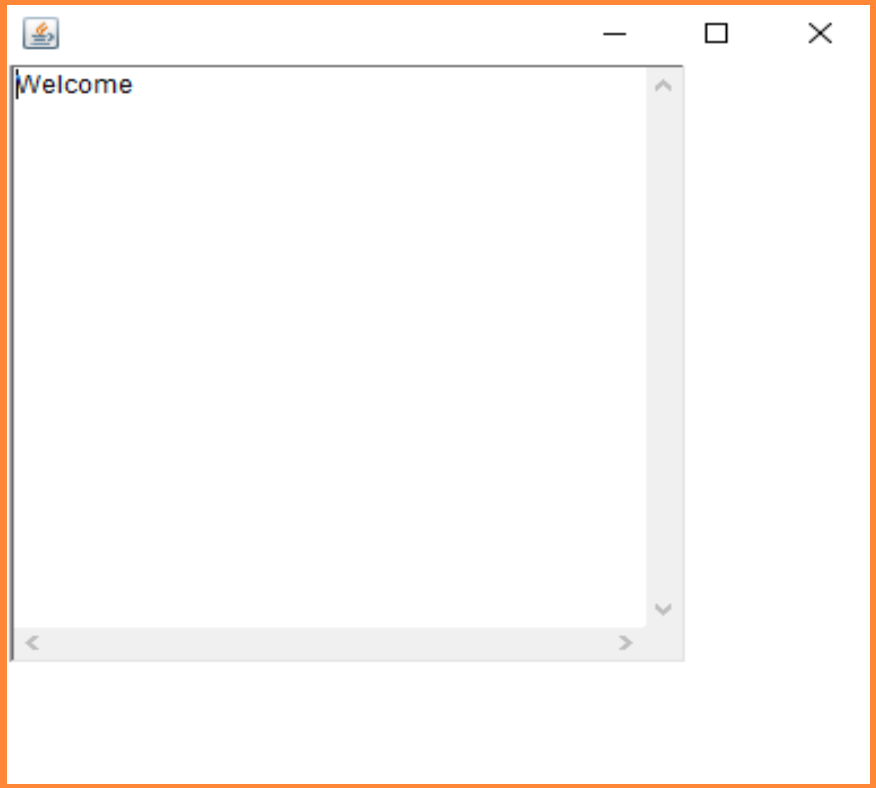
Java AWT TextArea

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

EXAMPLE-TEXTAREA

```
import java.awt.*;
public class First
{
    First(){
       Frame f= new Frame();
          TextArea area=new TextArea("Welcome");
       area.setBounds(10,30, 300,300);
       f.add(area);
       f.setSize(400,400);
       f.setLayout(null);
       f.setVisible(true);
    }
public static void main(String args[])
{
   new First();
}
}
```

OUTPUT



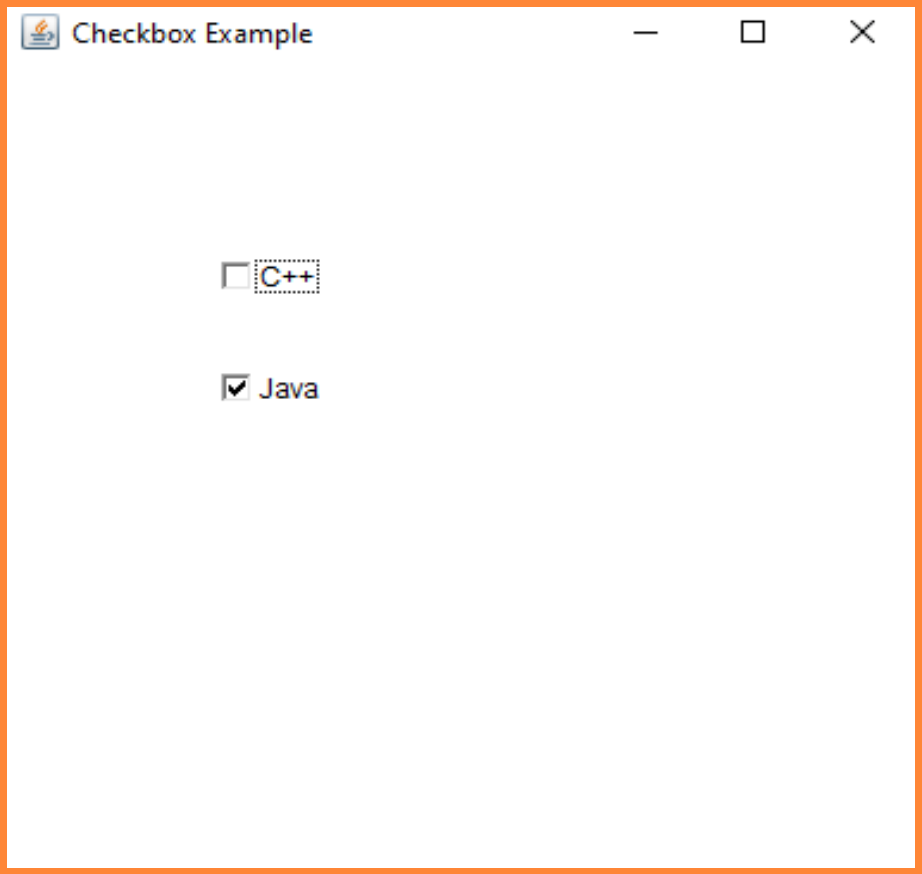
Java AWT Checkbox

- The Checkbox 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".

EXAMPLE-CHECKBOX

```
import java.awt.*;
public class First
{
    First(){
      Frame f= new Frame("Checkbox Example");
       Checkbox checkbox1 = new Checkbox("C++");
       checkbox1.setBounds(100,100, 50,50);
       Checkbox checkbox2 = new Checkbox("Java", true); //marking as true by
using second //argument
       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 First();
}
}
```

OUTPUT



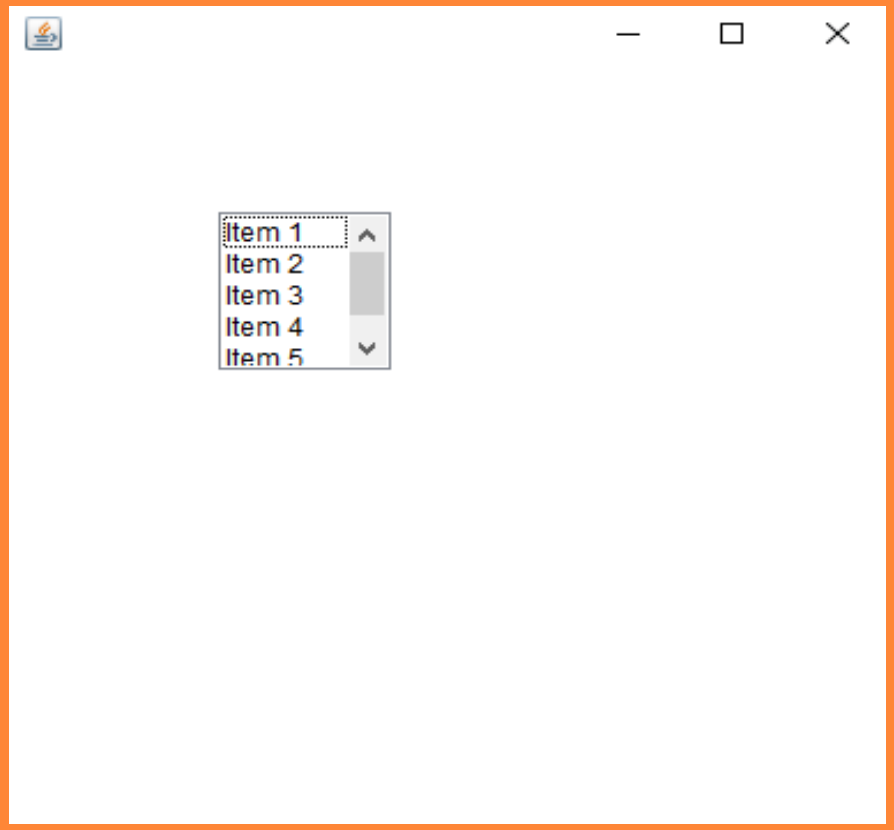
Java AWT List

- The object of List class represents a list of text items.
- By the help of list, user can choose either one item or multiple items.

EXAMPLE-LIST

```
import java.awt.*;
public class First
{
    First(){
       Frame f= new Frame();
       List l1=new List(5);
       l1.setBounds(100,100, 75,75);
       l1.add("Item 1");
       l1.add("Item 2");
       l1.add("Item 3");
       l1.add("Item 4");
       l1.add("Item 5");
       f.add(l1);
       f.setSize(400,400);
       f.setLayout(null);
       f.setVisible(true);
    }
public static void main(String args[])
{
   new First();
}
}
```

OUTPUT



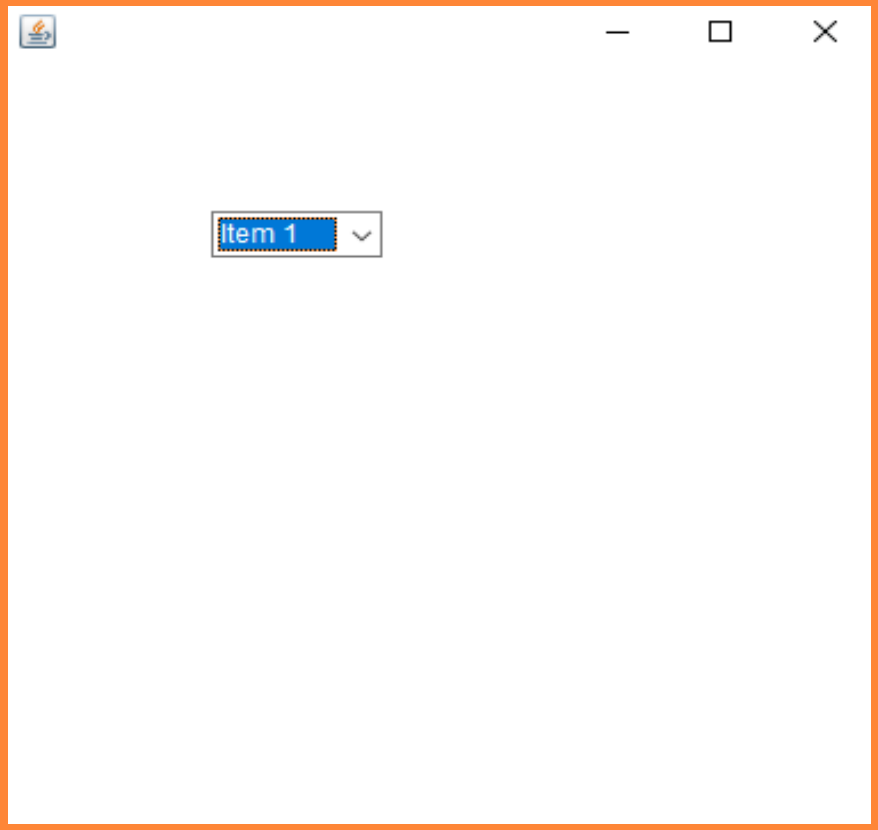
Java AWT Choice

- 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.

EXAMPLE-CHOICE

```
import java.awt.*;
public class Example
{
        Example(){
        Frame f= new Frame();
        Choice c=new Choice();
        c.setBounds(100,100, 75,75);
        c.add("Item 1");
        c.add("Item 2");
        c.add("Item 3");
        c.add("Item 4");
        c.add("Item 5");
        f.add(c);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
public static void main(String args[])
{
   new Example();
}
}
```

OUTPUT



Java AWT Scrollbar

- The object of Scrollbar class is used to add horizontal and vertical scrollbar.
- Scrollbar is a GUI component allows us to see invisible number of rows and columns.

EXAMPLE-SCROLLBAR

```
import java.awt.*;
class Example{
Example(){
        Frame f= new Frame("Scrollbar Example");
        Scrollbar s=new Scrollbar();
        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 Example();
}
}
```

OUTPUT