Applets & AWT Components

- Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

➢ Advantage of Applet

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many platforms, including Linux, Windows, Mac OS etc.

➢ Drawback of Applet

- Plugin is required at client browser to execute applet.

Lifecycle of Java Applet

- Applet is initialized.
- Applet is started.
- Applet is painted.
- Applet is stopped.
- Applet is destroyed.

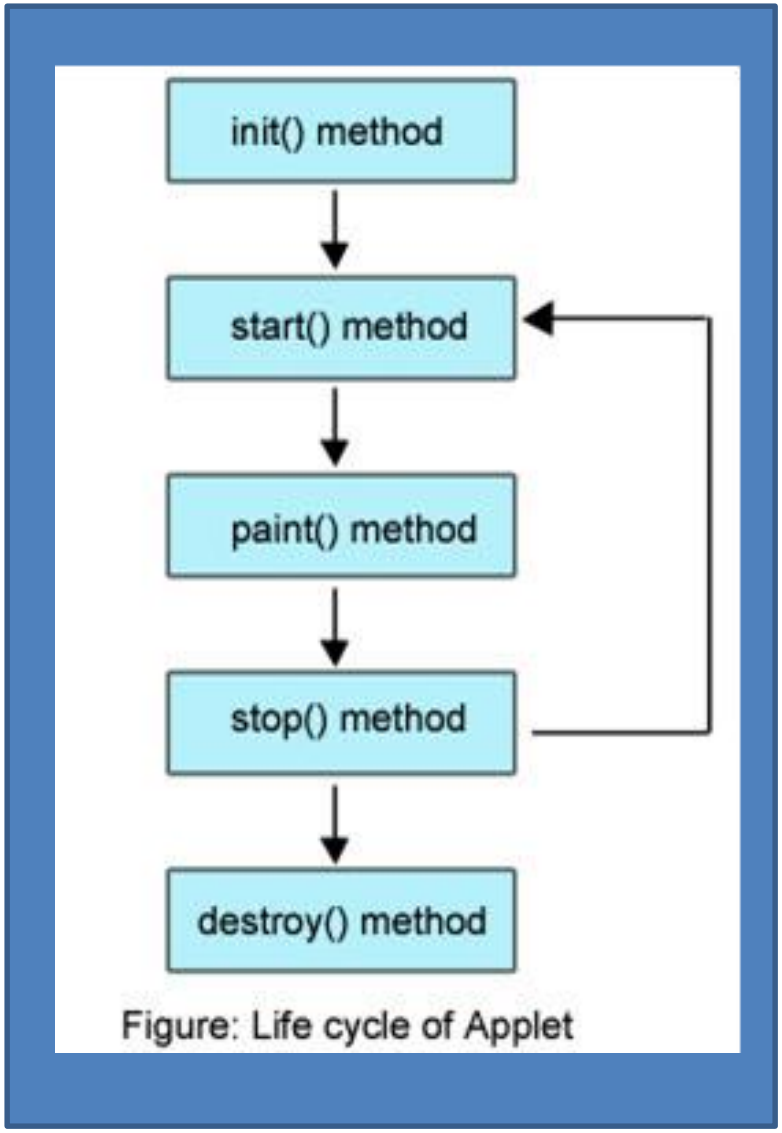


Figure: Life cycle of Applet

Lifecycle methods for Applet:

- The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides life cycle methods for an applet.
- java.applet.Applet class for creating any applet java.applet.Applet class must be inherited.
- It provides 4 life cycle methods of applet.
- **public void init():** is used to initialized the Applet. It is invoked only once.
- **public void start():** is invoked after the init() method or browser is maximized. It is used to start the Applet.
- **public void stop():** is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
- **public void destroy():** is used to destroy the Applet. It is invoked only once.

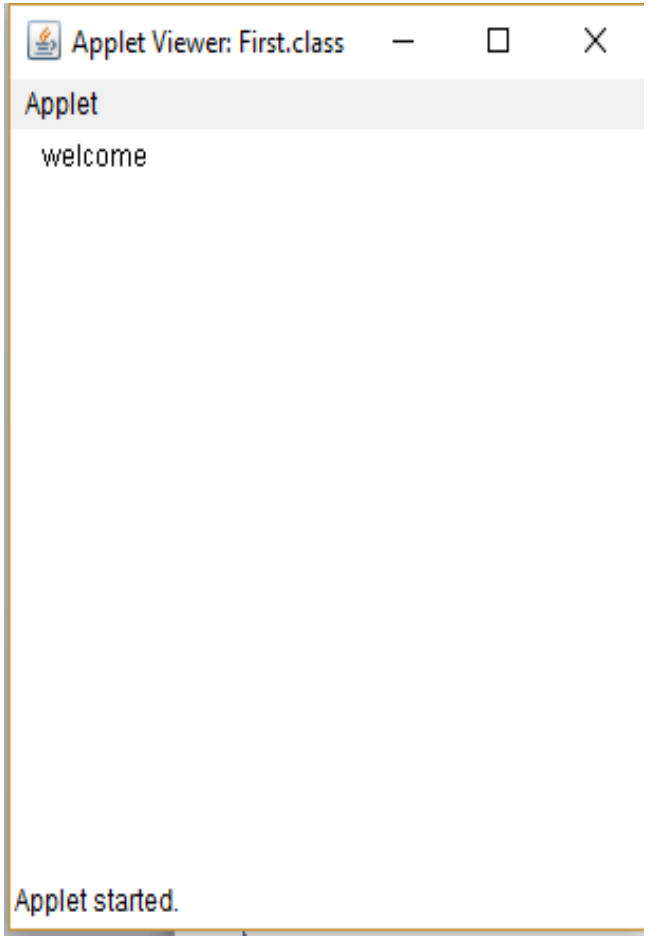
- java.awt.Component class
- The Component class provides 1 life cycle method of applet.
- **public void paint(Graphics g):** is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

- Who is responsible to manage the life cycle of an applet?
- Java Plug-in software.
- How to run an Applet?
- There are two ways to run an applet
- By html file.
- By appletviewer tool

Simple example of Applet by appletviewer tool:

```
//First.java
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet
{
public void paint(Graphics g)
{
g.drawString("welcome",150,150);
}
}
/*<applet code="First.class" width="300" height="300">
</applet>
*/
```

```
D:\JAVAPROGRAMMING\Applets>javac First.java

D:\JAVAPROGRAMMING\Applets>appletviewer First.java
```



- To execute the applet by appletviewer tool, write in command prompt:
- **c:\>**javac First.java
- **c:\>**appletviewer First.java

Displaying Graphics in Applet

- java.awt.Graphics class provides many methods for graphics programming.

➢ Commonly used methods of Graphics class:

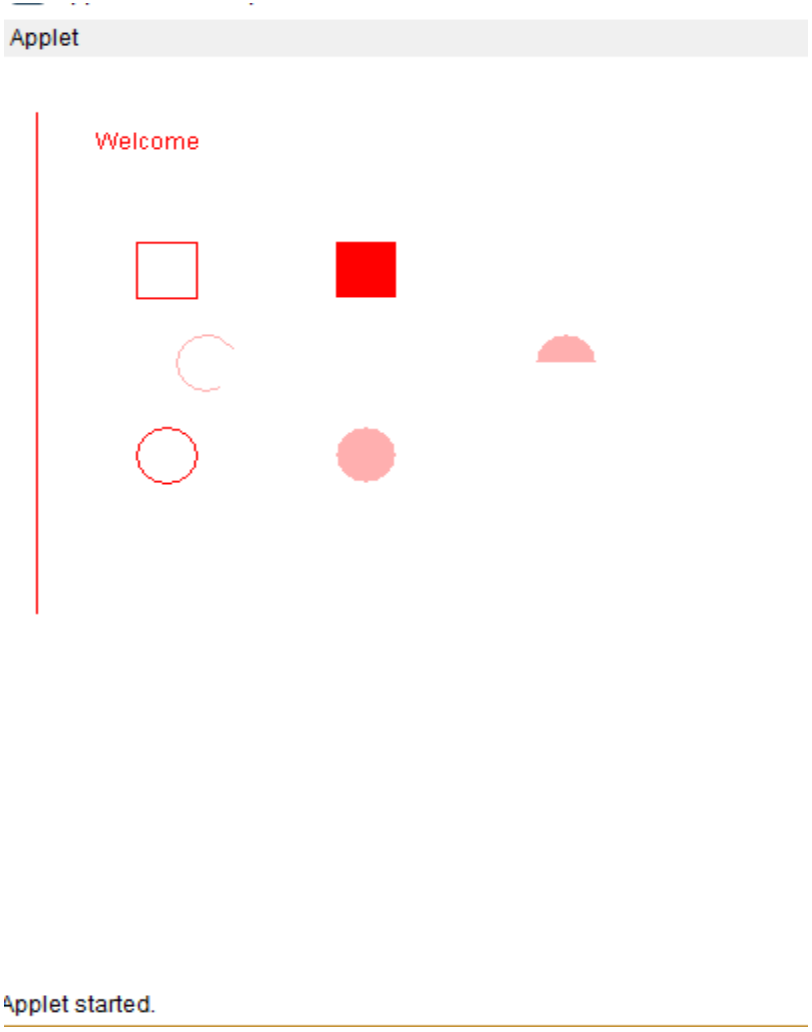
- **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
- **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
- **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
- **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
- **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.

- **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).
- **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
- **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
- **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):**is used to fill a circular or elliptical arc.
- **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
- **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

Example of Graphics in applet:

```
import java.applet.Applet;
import java.awt.*;
public class GraphicsDemo extends Applet
{
public void paint(Graphics g){
g.setColor(Color.red);
g.drawString("Welcome",50, 50);
g.drawLine(20,30,20,300);
g.drawRect(70,100,30,30);
g.fillRect(170,100,30,30);
g.drawOval(70,200,30,30);
g.setColor(Color.pink);
g.fillOval(170,200,30,30);
g.drawArc(90,150,30,30,30,270);
g.fillArc(270,150,30,30,0,180);
}
}
```

```
D:\JAVAPROGRAMMING\Applets>javac GraphicsDemo.java

D:\JAVAPROGRAMMING\Applets>appletviewer GraphicsDemo.java
```



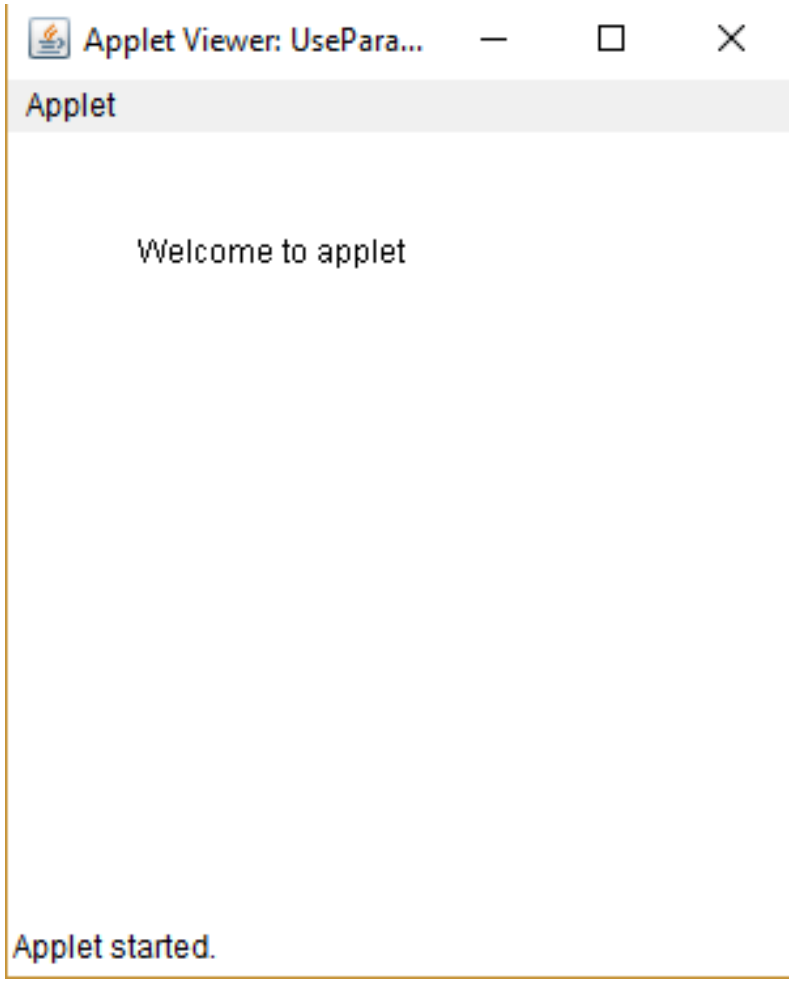
Parameter passing in Applet

- We can get any information from the HTML file as a parameter. For this purpose, Applet class provides a method named getParameter().
- Syntax:
- **public** String getParameter(String parameterName)

Example of using parameter in Applet:

```
import java.applet.Applet;
import java.awt.Graphics;
public class UseParam extends Applet{
public void paint(Graphics g){
String str=getParameter("msg");
g.drawString(str,50, 50);
}
}
/*<applet code="UseParam.class" width="300" height="300">
<param name="msg" value="Welcome to applet">
</applet>*/
```

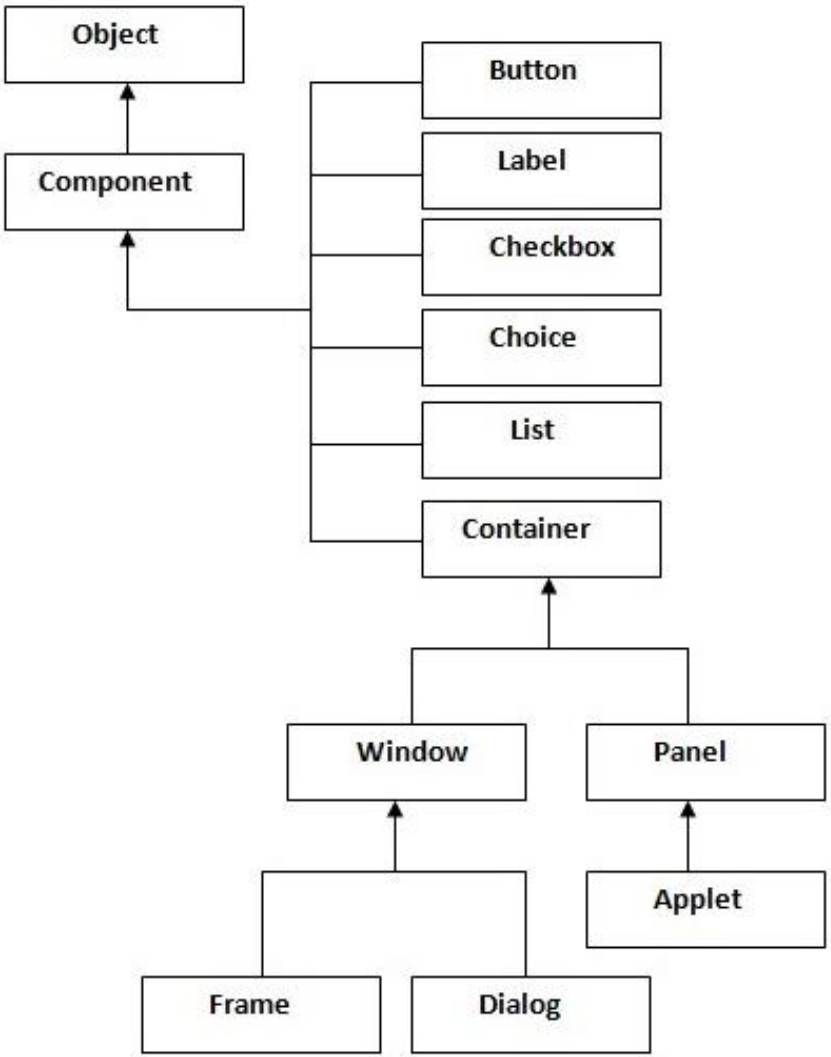
```
D:\JAVAPROGRAMMING\Applets>javac GraphicsDemo.java

D:\JAVAPROGRAMMING\Applets>appletviewer GraphicsDemo.java
```



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.

Java AWT Hierarchy



➢ 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.

➢ 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 Component 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.

AWT Example by Inheritance

```
import java.awt.*;
class First extends Frame{
First(){
Button b=new Button("click me");
b.setBounds(30,100,80,30);// setting button position
add(b);//adding button into frame
setSize(300,300);//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();
}
}
```

Event and Listener (Java Event Handling)

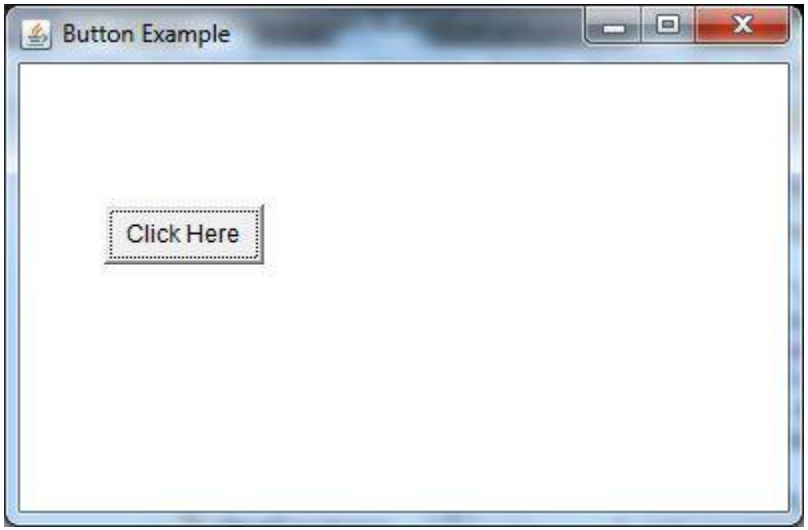
Changing the state of an object is known as an event.
For example, click on button, dragging mouse etc.
The java.awt.event package provides many event classes and Listener interfaces for event handling.

Java Event classes and Listener interfaces

Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener

Java AWT Button

```
import java.awt.*;
public class ButtonExample {
public static void main(String[] args) {
    Frame f=new Frame("Button Example");
    Button b=new Button("Click Here");
    b.setBounds(50,100,80,30);
    f.add(b);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

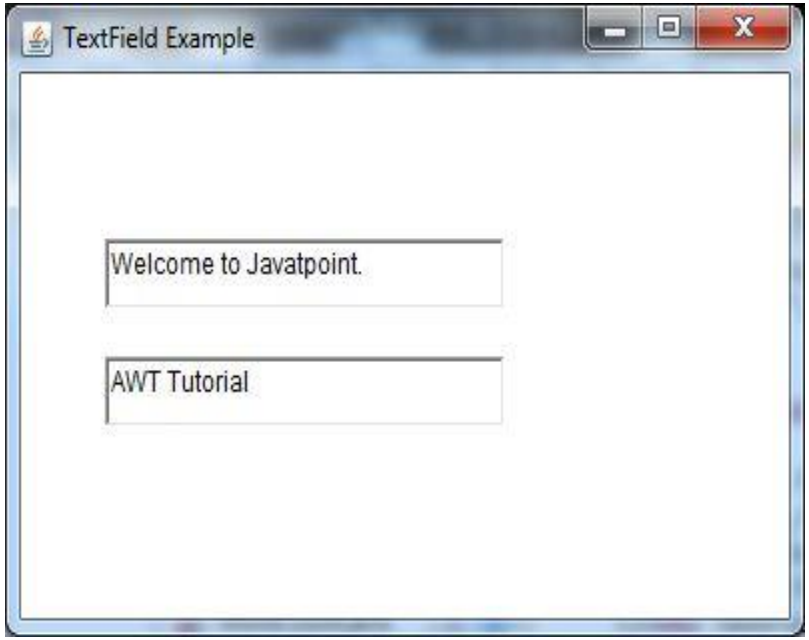


Java Label Example

```
import java.awt.*;
class LabelExample{
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);
}
}
```

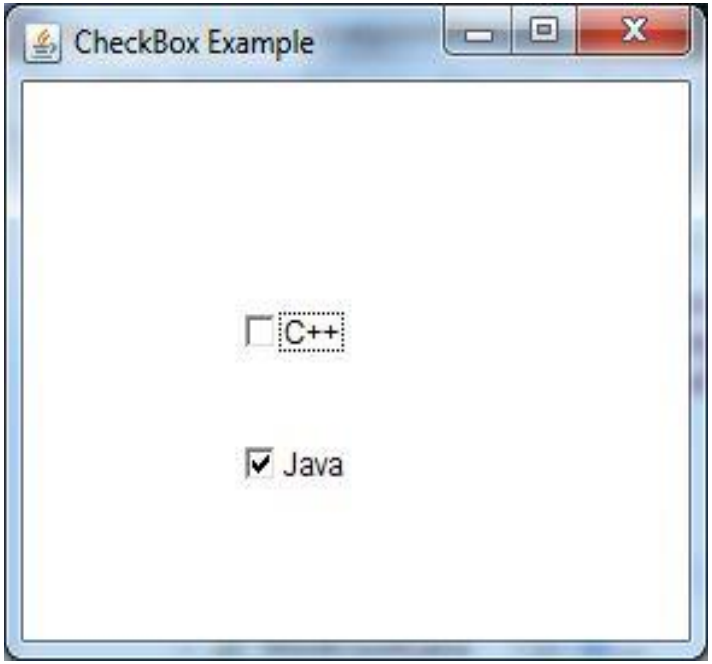
Java AWT TextField Example

```
import java.awt.*;
class TextFieldExample{
public static void main(String args[]){
    Frame f= new Frame("TextField Example");
    TextField t1,t2;
    t1=new TextField("Welcome to Javatpoint.");
    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);
}
}
```



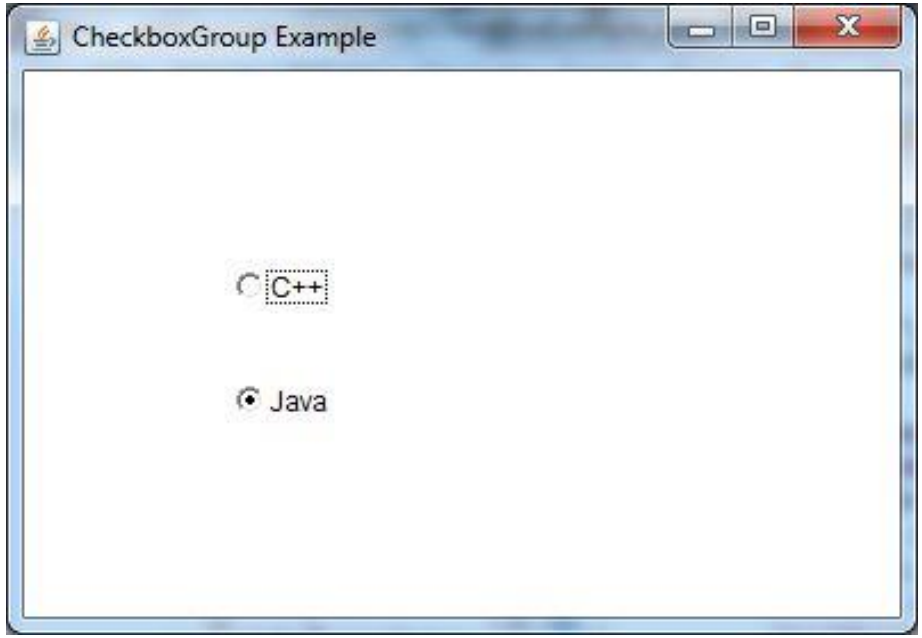
Java AWT Checkbox Example

```
import java.awt.*;
public class CheckboxExample
{
    CheckboxExample(){
      Frame f= new Frame("Checkbox Example");
       Checkbox checkbox1 = new Checkbox("C++");
       checkbox1.setBounds(100,100, 50,50);
       Checkbox checkbox2 = new Checkbox("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();
}
}
```



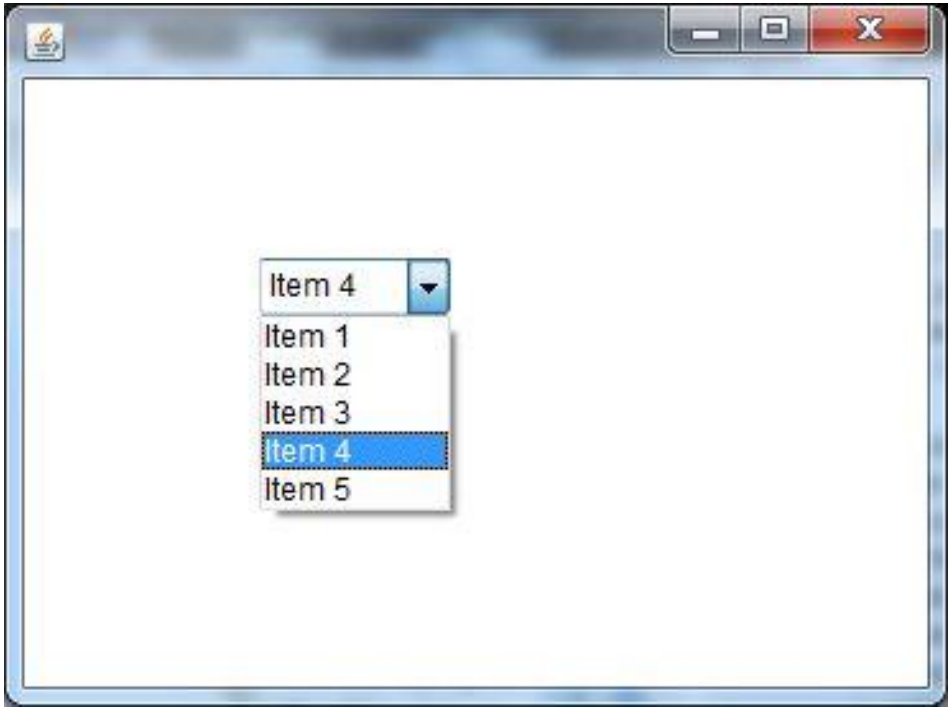
Java AWT CheckboxGroup Example

```
import java.awt.*;
public class CheckboxGroupExample
{
     CheckboxGroupExample(){
     Frame f= new Frame("CheckboxGroup Example");
      CheckboxGroup cbg = new CheckboxGroup();
      Checkbox checkBox1 = new Checkbox("C++", cbg, false);
      checkBox1.setBounds(100,100, 50,50);
      Checkbox checkBox2 = new Checkbox("Java", cbg, 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 CheckboxGroupExample();
}
}
```



Java AWT Choice Example

```
import java.awt.*;
public class ChoiceExample
{
      ChoiceExample(){
      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 ChoiceExample();
}
}
```



Java AWT List Example

```
import java.awt.*;
public class ListExample
{
    ListExample(){
       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 ListExample();
}
}
```