

* AWT Controls:-

=====

Every AWT Controls inherits Properties from Component class

Control	Description
Label	A Label object is a component for placing text in a container.
Button	This class creates a button.
Check Box	A checkbox is a graphical component that can be in either an on or off.
TextBox	A TextField object is a text component that allows for editing of single line of text.
TextArea	A TextArea object is a text component that allows for editing of multiple lines of text.
RadioButton	It is special kind of checkbox that is used select only one option.
Choice	A Choice control is used to show popup menu of choices.
List	It presents the user with scrolling list of ^{text} items.

① AWT Label:-

- In Java, AWT Contains a label class
- It is used for placing text in a container.
- Only single line text is allowed.
- It does not create any event.
- Label control is object of label.

Creating Label:- `Label l = new Label("String");`

Label Methods:-

1. `void setText(String str)`
2. `getText()`
3. `setAlignment()` // Set the alignment
4. `getAlignment()` // Obtain the current alignment

Example:- `Label l = new Label("JPweb");`
`l.setText("JPnotes");` // Set or change text
`l.getText()` // Returns string value

```
import java.awt.*;
```

```
class LabelDemo
```

```
{
```

```
    public static void main (String args [])
```

```
    {
```

```
        Frame f = new Frame("JPnotes");
```

```
        Label l = new Label("Free notes of java");
```

```
        l.setBounds(50, 50, 200, 50);
```

```
        f.add(l);
```

f.setSize (500, 500);

f.setVisible (true);

}

}

② AWT Buttons :-

- The most widely used control is Button.
- It is used for creating a labelled button which can perform an action.

Creating Button = Button b = new Button (String label);

Example :-

```
import java.awt.*;  
public class ButtonDemo  
{
```

```
public static void main (String args[])  
{
```

```
Frame f = new Frame ("jnotes");
```

```
Button b = new Button ("click here");
```

```
b.setBounds (50, 100, 200, 30);
```

```
f.add (b);
```

```
f.setSize (500, 500);
```

```
f.setLayout (null);
```

```
f.setVisible (true);
```

```
}
```

```
}
```

③. AWT TextField :-

In Java, AWT contains a TextField class, It is used for displaying a single line Text.

(Creating Label :-) `TextField txt = new TextField(25);`

→ if we want to show the text for the TextField :-

`TextField txt = new TextField("full name");`

→ TextField Methods :-

Set Text()

get Text()

Example :-

```
import java.awt.*;  
class TextFieldExample  
{
```

```
    public static void main (String args[]) {
```

```
        Frame f = new Frame("jweb");
```

```
        TextField t1 = new TextField("Welcome to jnotes");
```

```
        t1.setBounds(50, 100, 200, 30);
```

```
        f.add(t1);
```

```
        f.setSize(500, 500);
```

```
        f.setLayout(null);
```

```
        f.setVisible(true);
```

```
    }
```

```
}
```

④ AWT TextArea :-

The TextArea class provides a Component that displays multiple lines of text and allows the user to edit the text.

Difference between the TextField and TextArea :-

TextField :- allows single line of text.

TextArea :- allows multiple lines of text.

Syntax :- `TextArea t = new TextArea("Enter Address");`

Example :-

```
import java.awt.*;
public class TArea
{
    TArea()
    {
        Frame f = new Frame();
        TextArea t = new TextArea("Enter Address");
        t.setBounds(30, 40, 200, 200);
        f.add(t);
        f.setSize(500, 500);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main (String args[]) {
        new TArea();
    }
}
```

3 3


```

Checkbox cb = new Checkbox("Notes", true);
Checkbox cb2 = new Checkbox("Programs");
cb.setBounds(100, 100, 60, 60);
cb2.setBounds(100, 150, 60, 60);
f.add(cb);
f.add(cb2);
f.setSize(500, 500);
f.setLayout(null);
f.setVisible(true);
}

public static void main (String args [])
{
    CheckboxDemo Demo = new CheckboxDemo();
}
}

```

⑥ RADIO BUTTONS:-

- In AWT, there is no separate class for creating radio buttons.
- In radiobutton, a user can select only one radio button in a group.
- It is created by using Checkbox class and CheckboxGroup class.
- When checkboxes are grouped then only one box

can be checked at a time.

Note:- At a time only one checkbox button is allowed to be in "on" state and remaining checkbox button in "off" state.

Syntax:- `CheckboxGroup cbg = new CheckboxGroup();`

Example:-

```
import java.awt.*;  
public class Demo  
{  
    Demo()  
{  
    Frame f = new Frame("jprwebdevelopers");  
    CheckboxGroup cbg = new CheckboxGroup();  
    Checkbox cb1 = new Checkbox("male", cbg, false);  
    Checkbox cb2 = new Checkbox("female", cbg, false);  
    cb1.setBounds(100, 100, 50, 50);  
    cb2.setBounds(100, 150, 50, 50);  
    f.add(cb1);  
    f.add(cb2);  
    f.setSize(500, 500);  
    f.setLayout(null);  
    f.setVisible(true);  
}
```

}

```
public static void main(String args[]) {  
    Demo d = new Demo();  
}
```

}

}

⊕ AWT Choice :-

- Choice class is a part of Java AWT.
- The Choice class presents a pop-up menu for the user, the user may select an item from pop up menu.
- The selected item appears on top.

Constructors:-

Choice() :- Creates a new empty choice menu.

Methods:-

void add It adds an item to the choice menu.

String getItem(int index) It get the item (String) at the given index position in the choice menu.

int getItemCount() It returns the number of items of the choice menu.

void insert(String item, int index) :- Insert the item into the choice at the specified position.

void remove(int position) :- Removes the item from the choice menu at the given index position.

void removeAll() :- It removes all the items from the choice menu.

void select (int index): - To select an item based on the ^{given} index.
 void select (String name): - To select an item based on item name
 void getItem (int index): - To retrieve an item at given index.

Example

```

import java.awt.*;
public class ChoiceDemo
{
    ChoiceDemo ()
    {
        Frame f = new Frame();
        Choice c = new Choice();
        c.setBounds (100, 100, 75, 75);
        c.add ("Notes");
        c.add ("PPT");
        c.add ("Programs");
        c.add ("Projects");
        f.add (c);
        f.setSize (400, 400);
        f.setLayout (null);
        f.setVisible (true);
    }
    public static void main (String args[])
    {
        ChoiceDemo cdemo = new ChoiceDemo();
    }
}

```

8 AWT LIST :-

- The object of list class represents a list of text items.

- With the help of the list class, user can choose either one item or multiple items.

→ List l1 = new List (5);
List (int row_num) list of 5 rows.

Methods:-

- void add (String item) → It adds the item.
- void add (String item, int index) - adds the item into list at the given index position.
- int getItemCount () :- It gets the count of items in the list.
- int getRows () It fetches the count of visible rows in the list.
- getSelectedItems () :- It gets the selected items on the list.
- void removeAll () It removes all the items from the list.

//_

Example :-

```
import java.awt.*;
```

```
public class ListDemo  
{
```

```
    ListDemo()  
{
```

```
        Frame f = new Frame ();
```

```
        List l1 = new List ();
```

```
        l1.setBounds (100, 100, 75, 75);
```

```
        l1.add ("notes");
```

```
        l1.add ("ppt");
```

```
        l1.add ("programs");
```

```
        l1.add ("projects");
```

```
        f.add (l1);
```

```
        f.setSize (400, 400);
```

```
        f.setLayout (null);
```

```
        f.setVisible (true);
```

```
    }
```

```
    public static void main (String args [])
```

```
    {
```

```
        ListDemo ld = new ListDemo ();
```

```
    }
```

```
}
```

Handwritten notes by Ipwebdevelopers