**Applet**

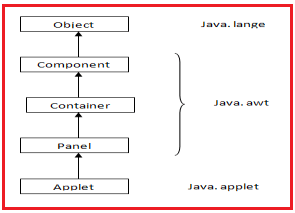
**What is AWT?**

* **AWT (Abstract Window Toolkit)** is a part of Java's standard library used for creating graphical user interfaces (GUI).
* It provides a collection of pre-built classes and interfaces for building platform-independent GUIs.
* AWT components are **heavyweight**, meaning they rely on the native GUI components of the underlying operating system.

**Features of AWT**

1. **Platform Independence**
   * AWT ensures that GUIs can run on any platform where Java is installed.
   * It uses the **native look and feel** of the operating system.
2. **Event-Driven Programming**
   * AWT follows an **event-driven model**, where user interactions (like button clicks or mouse movements) generate events that can be handled using listeners.
3. **Rich GUI Components**
   * AWT provides various GUI components, such as buttons, text fields, checkboxes, labels, menus, etc.
4. **Layout Management**
   * It supports layout managers like **FlowLayout**, **BorderLayout**, **GridLayout**, etc., to arrange components.

**AWT Hierarchy**



The AWT library is organized in a hierarchical structure. The root of the hierarchy is the Component class. Here's a simplified view:

**1. Core Classes**

* **Component**  
  The base class for all AWT components (e.g., Button, Label, TextField).
* **Container**  
  A subclass of Component that can hold and organize other components. Examples: Frame, Panel.

**2. Common Components**

* **Label**: Displays a single line of uneditable text.
* **Button**: A clickable button for performing actions.
* **TextField**: Allows the user to input a single line of text.
* **Checkbox**: A box that can be checked or unchecked.
* **List**: Displays a list of items from which the user can select.

**3. Containers**

* **Panel**: A generic container for organizing components.
* **Frame**: Represents the main window of an application.
* **Dialog**: A pop-up window used for temporary interactions with the user.

**4. Layout Managers**

* **FlowLayout**: Arranges components in a left-to-right flow.
* **BorderLayout**: Divides the container into five regions (North, South, East, West, Center).
* **GridLayout**: Arranges components in a grid of rows and columns.

**Common AWT Classes**

1. **Frame**
   * A top-level window with a title bar, close/minimize buttons, and a content area.
   * Example:

import java.awt.\*;

public class MyFrame {

public static void main(String[] args) {

Frame frame = new Frame("My AWT Frame");

frame.setSize(400, 300);

frame.setVisible(true);

}

}

**Button**

* Used to create a clickable button.
* Example:

import java.awt.\*;

public class MyButton {

public static void main(String[] args) {

Frame frame = new Frame("Button Example");

Button button = new Button("Click Me");

button.setBounds(50, 50, 80, 30); // x, y, width, height

frame.add(button);

frame.setSize(300, 200);

frame.setLayout(null);

frame.setVisible(true);

}

}

**Label**

* Displays a piece of text.
* Example:

import java.awt.\*;

public class MyLabel {

public static void main(String[] args) {

Frame frame = new Frame("Label Example");

Label label = new Label("Hello, AWT!");

label.setBounds(50, 50, 100, 30); // x, y, width, height

frame.add(label);

frame.setSize(300, 200);

frame.setLayout(null);

frame.setVisible(true);

}

}

**Event Handling in AWT**

AWT components generate **events** based on user actions. These events are handled using **event listeners**.

**Steps for Event Handling:**

1. Implement a listener interface (e.g., ActionListener, MouseListener).
2. Register the listener with the component.
3. Override the appropriate event-handling methods.

**Example: Handling Button Click**

import java.awt.\*;

import java.awt.event.\*;

public class ButtonClickExample {

public static void main(String[] args) {

Frame frame = new Frame("Event Handling Example");

Button button = new Button("Click Me");

button.setBounds(50, 50, 80, 30); // x, y, width, height

frame.add(button);

// Add ActionListener to the button

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.out.println("Button clicked!");

}

});

frame.setSize(300, 200);

frame.setLayout(null);

frame.setVisible(true);

}

}

Java program that includes **Label**, **Button**, **TextField**, **Checkbox**, and **List**, with basic functionality:

Program: AWT Components Example

**import java.awt.\*;**

**import java.awt.event.\*;**

**public class AWTComponentsExample {**

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

**// Create a Frame**

**Frame frame = new Frame("AWT Components Example");**

**// Set Frame Layout**

**frame.setLayout(null);**

**frame.setSize(400, 400);**

**// Label**

**Label label = new Label("Enter your name:");**

**label.setBounds(50, 50, 120, 30);**

**frame.add(label);**

**// TextField**

**TextField textField = new TextField();**

**textField.setBounds(180, 50, 150, 30);**

**frame.add(textField);**

**// Checkbox**

**Checkbox checkbox = new Checkbox("Subscribe to Newsletter");**

**checkbox.setBounds(50, 100, 200, 30);**

**frame.add(checkbox);**

**// List**

**Label listLabel = new Label("Select a language:");**

**listLabel.setBounds(50, 150, 120, 30);**

**frame.add(listLabel);**

**List languageList = new List(4, false);**

**languageList.setBounds(50, 180, 150, 80);**

**languageList.add("Java");**

**languageList.add("Python");**

**languageList.add("C++");**

**languageList.add("JavaScript");**

**frame.add(languageList);**

**// Button**

**Button button = new Button("Submit");**

**button.setBounds(50, 280, 80, 30);**

**frame.add(button);**

**// Event Handling for Button Click**

**button.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**// Collect data**

**String name = textField.getText();**

**boolean subscribed = checkbox.getState();**

**String language = languageList.getSelectedItem();**

**// Display the collected data in the console**

**System.out.println("Name: " + name);**

**System.out.println("Subscribed: " + (subscribed ? "Yes" : "No"));**

**System.out.println("Language: " + language);**

**}**

**});**

**// Close the frame on clicking the close button**

**frame.addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e) {**

**frame.dispose();**

**}**

**});**

**// Make Frame visible**

**frame.setVisible(true);**

**}**

**}**