Bhaskar Joshi

Java AWT (Abstract Window Toolkit) - Complete Guide

1 What is AWT?

AWT (**Abstract Window Toolkit**) is Java's **built-in** GUI (Graphical User Interface) library that provides **components like Buttons, TextFields, Labels, Frames, etc.** for creating **desktop applications**.

✅ Key Features of AWT:

✔ AWT is **part of Java** (java.awt package).
✔ It is **platform-dependent** (uses native OS components).
✔ Components are **heavyweight** (linked with OS UI).
✔ Provides basic **GUI controls**, layout managers, and event handling.

---

2 Basic Structure of an AWT Program

Steps to Create an AWT GUI Application:

1. **Create a Frame (Window)**
2. **Add Components (Button, Label, etc.)**
3. **Use Layout Manager to Arrange Components**
4. **Handle User Events (Button Click, Key Press, etc.)**
5. **Make Frame Visible**

---

3 Creating a Basic AWT Window

Here's the simplest **AWT program** to display a window with a button:

```
import java.awt.*;  // Import AWT package
import java.awt.event.*;  // Import for event handling

public class AWTExample extends Frame {
   AWTExample() {
      // Set Frame Title
      setTitle("My First AWT Window");

      // Set Size (Width, Height)
      setSize(400, 300);

      // Set Layout (Default: BorderLayout)
      setLayout(new FlowLayout());

      // Create a Button
```

```
        Button btn = new Button("Click Me");

        // Add Button to Frame
        add(btn);

        // Add Window Closing Event
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose(); // Close the window
            }
        });

        // Make Frame Visible
        setVisible(true);
    }

    public static void main(String[] args) {
        new AWTExample();  // Create and Show Window
    }
}
```

---

4 Understanding the Code

Code	Explanation
extends Frame	Inherits Frame to create a window
setTitle("My First AWT Window")	Sets window title
setSize(400, 300)	Sets width and height of the window
setLayout(new FlowLayout())	Arranges components in a row
Button btn = new Button("Click Me")	Creates a button
add(btn)	Adds button to the frame
setVisible(true)	Displays the window
addWindowListener(new WindowAdapter() {...})	Closes the window when clicking the close button

---

5 Adding More AWT Components

AWT provides multiple **components** to build a UI. Here's how to add some common ones:

Example: TextField, Label, and Button

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

public class AWTComponents extends Frame implements ActionListener {
    TextField tf;

    AWTComponents() {
        setTitle("AWT Components Example");
        setSize(400, 300);
```

```
        setLayout(new FlowLayout());

        Label label = new Label("Enter Name:");
        tf = new TextField(20);
        Button btn = new Button("Submit");

        btn.addActionListener(this); // Register event

        add(label);
        add(tf);
        add(btn);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });

        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        System.out.println("Submitted Name: " + tf.getText());
    }

    public static void main(String[] args) {
        new AWTComponents();
    }
}
```

---

6 Event Handling in AWT

- **Events** are actions like **button clicks, key presses, mouse clicks, etc.**
- To handle events, AWT uses **event listeners**.

Example: Button Click Event Handling

```
btn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Button Clicked!");
    }
});
```

Event Listener	Description
ActionListener	Handles button clicks
KeyListener	Handles keyboard input
MouseListener	Handles mouse clicks
WindowListener	Handles window events (closing, opening)

---

7 Layout Managers in AWT

AWT provides **layout managers** to arrange components inside a container.

Layout Manager	Description
FlowLayout	Places components **horizontally** (default)
BorderLayout	Divides container into **NORTH, SOUTH, EAST, WEST, CENTER**
GridLayout	Arranges components in **rows & columns**
CardLayout	Stacks components like **cards** (only one visible at a time)

Example: GridLayout

```
setLayout(new GridLayout(2, 2)); // 2 rows, 2 columns
```

---

8 Connecting AWT to a Database

You can connect AWT applications with databases (MySQL, PostgreSQL, etc.) using **JDBC**.

Example: AWT Form with MySQL Database

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class AWTDatabaseExample extends Frame implements ActionListener {
   TextField tf;
   Button btn;

   AWTDatabaseExample() {
      setTitle("Database Example");
      setSize(400, 300);
      setLayout(new FlowLayout());

      Label label = new Label("Enter Name:");
      tf = new TextField(20);
      btn = new Button("Save to DB");

      btn.addActionListener(this);

      add(label);
      add(tf);
      add(btn);

      addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent e) {
            dispose();
         }
      });

      setVisible(true);
   }

   public void actionPerformed(ActionEvent e) {
      String name = tf.getText();
```

```
        try {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root",
"password");
            PreparedStatement ps = con.prepareStatement("INSERT INTO users (name) VALUES (?)");
            ps.setString(1, name);
            ps.executeUpdate();
            con.close();
            System.out.println("Data Saved!");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new AWTDatabaseExample();
    }
}
```

✅ This connects to a **MySQL database**, inserts user input, and prints **"Data Saved!"**.

---

9 Modifying and Extending AWT Applications

- **Change the Look & Feel:** Use **Swing (**JFrame**,** JButton**)** for modern UI.
- **Use Custom Graphics:** Override paint(Graphics g) for **drawing shapes**.
- **Add File Handling:** Use FileDialog to **open/save files**.

Example: Drawing a Circle in AWT

```
import java.awt.*;

public class AWTGraphics extends Frame {
    AWTGraphics() {
        setTitle("AWT Graphics Example");
        setSize(400, 300);
        setVisible(true);
    }

    public void paint(Graphics g) {
        g.drawOval(100, 100, 50, 50); // Draw Circle
    }

    public static void main(String[] args) {
        new AWTGraphics();
    }
}
```

Conclusion

✔ AWT is Java's basic GUI toolkit for building desktop applications.
✔ It includes Frames, Buttons, TextFields, and other UI components.
✔ Events (like button clicks) are handled using event listeners.

✔ **AWT apps can be connected to databases (MySQL, PostgreSQL) via JDBC.**
✔ **For more advanced GUI, use** Swing **or** JavaFX**.**