Experiment No. 11
Implement a program on Applet or AWT Controls
Date of Performance:
Date of Submission:

Aim: Implement a program on Applet or AWT Controls

Objective:

To develop application like Calculator, Games, Animation using AWT Controls.

Theory:

Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applications in Java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavy weight i.e. its components are using the resources of underlying operating system (OS).

The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

1. A general interface between Java and the native system, used for windowing, events and layout managers. This API is at the core of Java GUI programming and is also used by Swing and Java 2D. It contains the interface between the native windowing system and the Java application1.

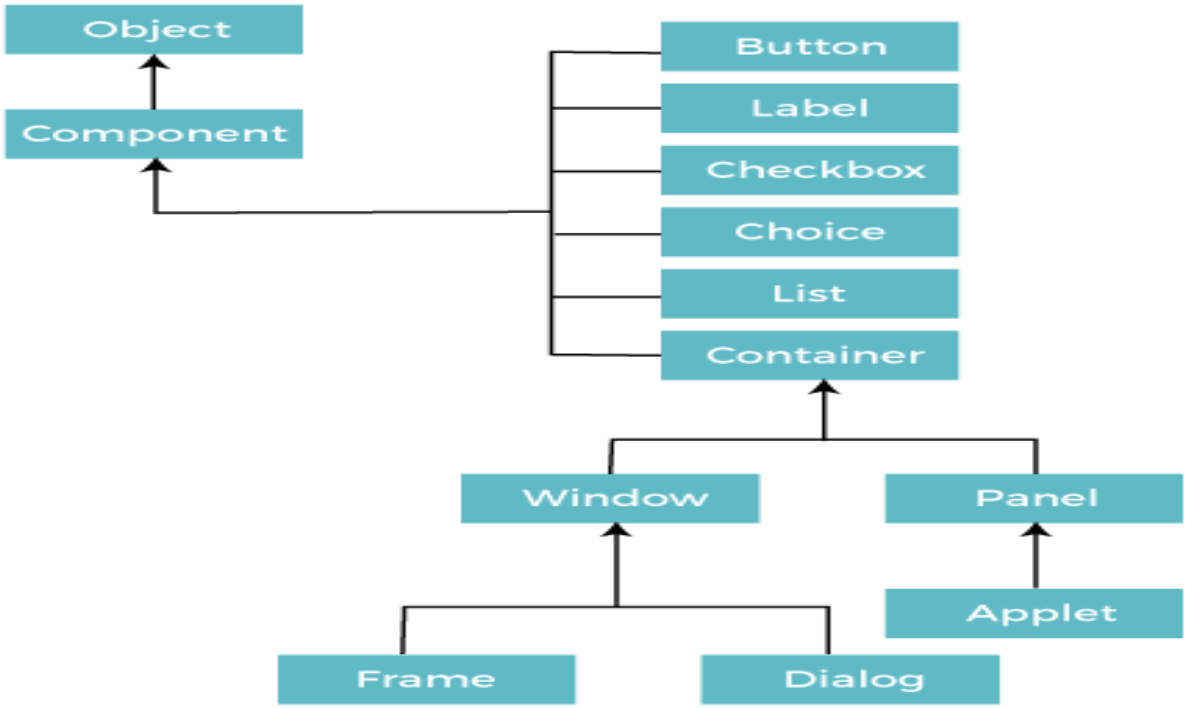
2. A basic set of GUI widgets such as buttons, text boxes, and menus1. AWT also provides Graphics and imaging tools, such as shape, color, and font classes2. AWT also avails layout managers which helps in increasing the flexibility of the window layouts2

Java AWT calls the native platform calls the native platform (operating systems) subroutine for creating API components like TextField, ChechBox, button, etc.

For example, an AWT GUI with components like TextField, label and button will have different look and feel for the different platforms like Windows, MAC OS, and Unix. The reason for this is the platforms have different view for their native components and AWT directly calls the native subroutine that creates those components.

In simple words, an AWT application will look like a windows application in Windows OS whereas it will look like a Mac application in the MAC OS.

Java AWT Hierarchy



Code:

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

public class AWTExample extends Frame implements ActionListener
{
   // Components of the GUI
   TextField textField;
   Button button;

   // Constructor
   public AWTExample()
{
```

```
    // Set up the Frame
    setTitle("AWT Example");
    setSize(300, 200);
    setLayout(new FlowLayout());

    // Create a TextField
    textField = new TextField(20);
    add(textField);

    // Create a Button
    button = new Button("Click Me");
    button.addActionListener(this);
    add(button);

    // Set the Frame visibility
    setVisible(true);

    // Close the application when the frame is closed
    addWindowListener(new WindowAdapter()
        {
        public void windowClosing(WindowEvent we)
        {
            System.exit(0);
        }
    });
}

// Action performed method
public void actionPerformed(ActionEvent e)
        {
    textField.setText("Button Clicked!");
}

// Main method
```

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

Conclusion:

While AWT provides a foundational understanding of GUI programming in Java, developers today are more likely to use Swing or JavaFX for new applications due to their enhanced capabilities and flexibility. However, AWT remains a useful tool for specific scenarios, particularly in learning contexts or for lightweight applications.