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.

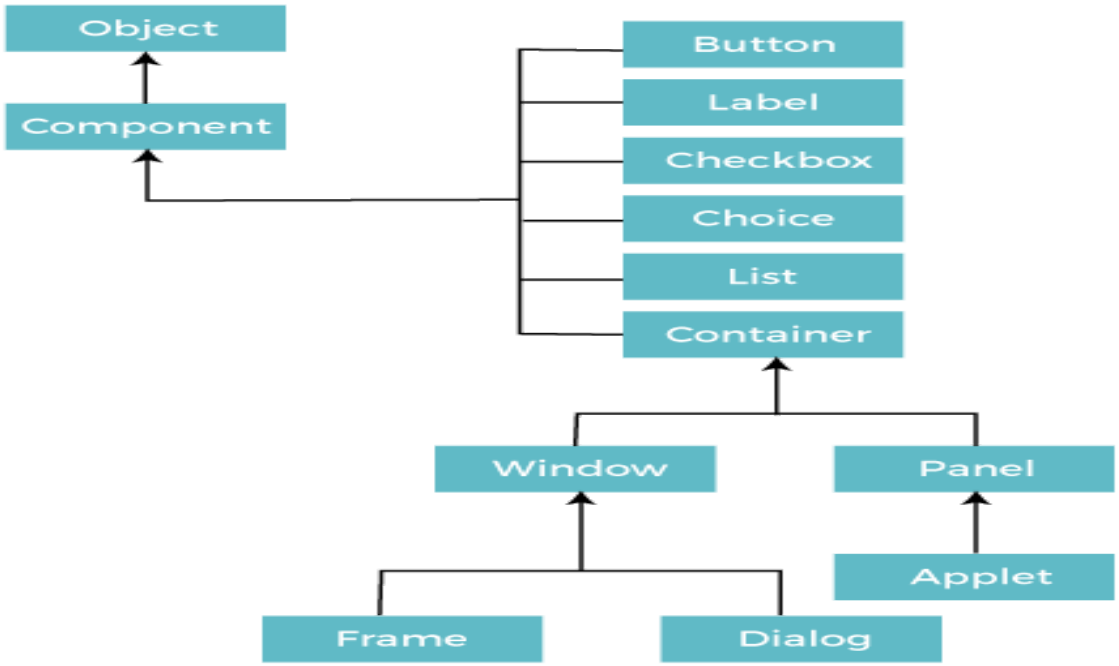
1. A general interface between Java and the native system, used for windowing, events and layout managers. This API is at the core of Java GUI programming and is also used by Swing and Java 2D. It contains the interface between the native windowing system and the Java application1.
2. A basic set of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

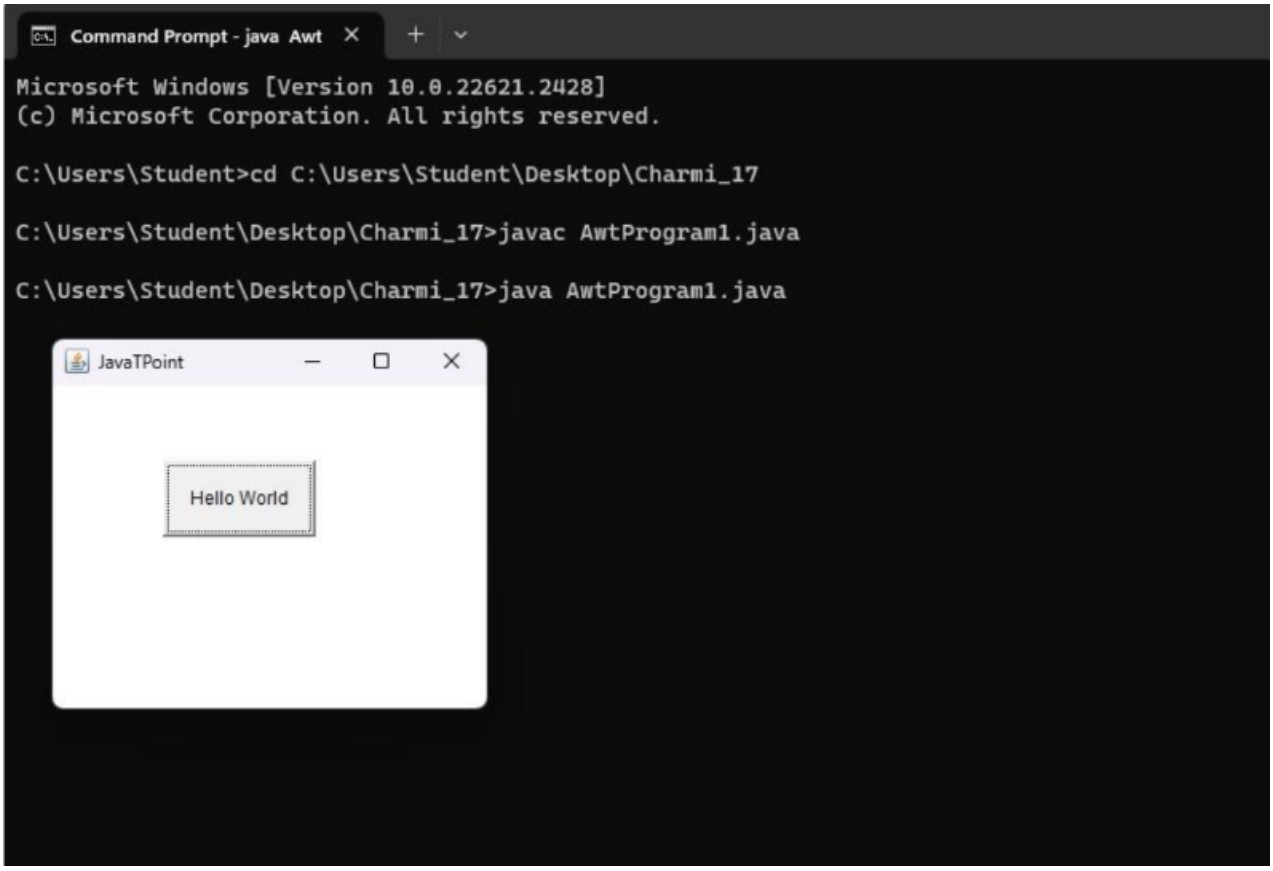
Java AWT Hierarchy



Code:

```
import java.awt.*;
public class AwtProgram1 {
public AwtProgram1()
{
Frame 1 = new Frame();
Button btn=new Button("Hello World");
btn.setBounds(80, 80, 100, 50);
f.add(btn);
f.setSize(300, 250);
f.setTitle("JavaTPoint");
f.setLayout(null);
f.setVisible(true);
}
public static void main(String[] args) {
```

```
AwtProgram1 awt = new AwtProgram1();
}
}
```



Conclusion:

Comment on application development using AWT Controls.

Utilizing AWT (Abstract Window Toolkit) controls for Java application development entails crafting graphical user interfaces (GUIs) tailored for desktop applications. AWT furnishes a fundamental assortment of GUI components, including buttons, labels, text fields, and more, forming the foundation of your application's interface. Here's a succinct overview:

1. AWT Controls: AWT equips you with an array of GUI controls to construct the user interface of your application.

2. Layout Managers: AWT introduces layout managers that assist in organizing and positioning these controls effectively within your GUI.

3. Customization: AWT grants you the liberty to fine-tune the visual attributes and interactive behavior of its controls.

4. Platform Independence: While AWT ensures platform independence, it may not deliver the most contemporary look and feel, depending on the underlying system.

5. Window and Frame: AWT empowers you to create top-level containers, such as the 'Frame,' serving as the principal windows for your application.