



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---

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



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

---

**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 application<sup>1</sup>.
2. A basic set of GUI widgets such as buttons, text boxes, and menus<sup>1</sup>. AWT also provides Graphics and imaging tools, such as shape, color, and font classes<sup>2</sup>. AWT also avails layout managers which helps in increasing the flexibility of the window layouts<sup>2</sup>

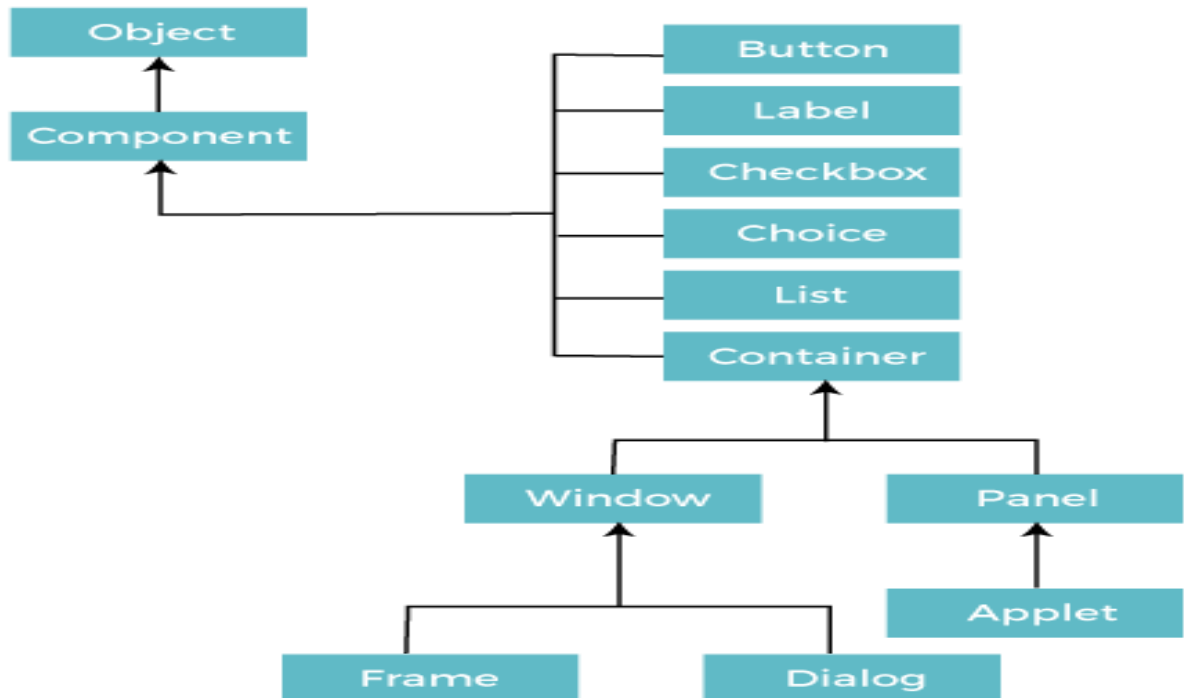
Java AWT calls the native platform calls the native platform (operating systems) subroutine for creating API components like TextField, ChechBox, button, etc.

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 f = 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();
```

```
}
```

```
}
```

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Student>cd C:\Users\Student\Desktop\Charmi_17

C:\Users\Student\Desktop\Charmi_17>javac AwtProgram1.java

C:\Users\Student\Desktop\Charmi_17>java AwtProgram1.java
```

### Conclusion:

Comment on application development using AWT Controls.

Application development using AWT (Abstract Window Toolkit) controls in Java provides a way to create graphical user interfaces (GUIs) for desktop applications. Here are key points regarding AWT controls in application development:

1. Cross-Platform Compatibility: AWT is part of the Java Standard Library and is designed to be platform-independent. This means that applications built using AWT controls can run on various operating systems without modification, making it a practical choice for cross-platform development.
2. Basic GUI Elements: AWT provides a set of basic GUI controls, including buttons, labels, text fields, checkboxes, radio buttons, and more. These controls allow developers to create interactive and user-friendly interfaces for their applications.



# **Vidyavardhini's College of Engineering and Technology**

## **Department of Artificial Intelligence & Data Science**

---

3. Event-Driven Programming: AWT applications are typically event-driven, meaning they respond to user interactions like button clicks and mouse movements. Event listeners and handlers are used to capture and respond to user actions, providing interactivity.

.