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:

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).

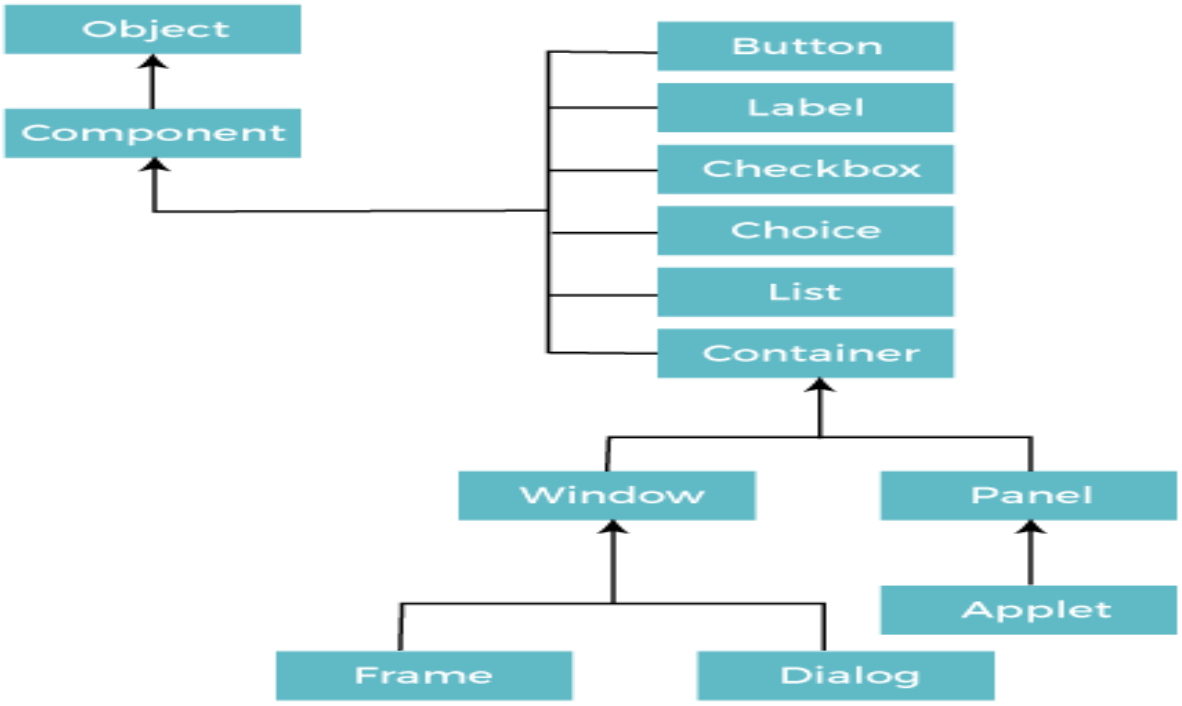
The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

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 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

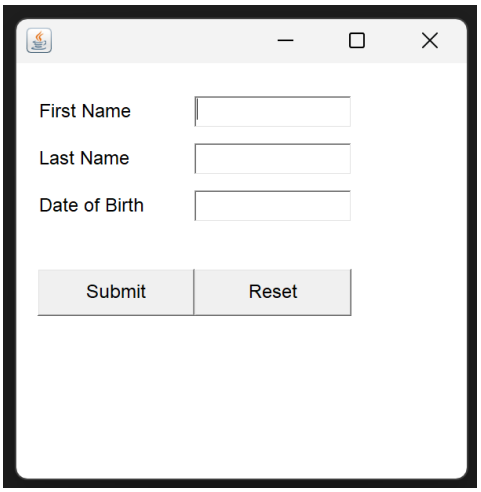


Code:

```
import java.awt.*;
public class AwtApp extends Frame {
AwtApp(){
Label firstName = new Label("First Name");
firstName.setBounds(20, 50, 80, 20);

Label lastName = new Label("Last Name");
```

```
lastName.setBounds(20, 80, 80, 20);

Label dob = new Label("Date of Birth");
dob.setBounds(20, 110, 80, 20);

TextField firstNameTF = new TextField();
firstNameTF.setBounds(120, 50, 100, 20);

TextField lastNameTF = new TextField();
lastNameTF.setBounds(120, 80, 100, 20);

TextField dobTF = new TextField();
dobTF.setBounds(120, 110, 100, 20);

Button sbmt = new Button("Submit");
sbmt.setBounds(20, 160, 100, 30);

Button reset = new Button("Reset");
reset.setBounds(120,160,100,30);

add(firstName);
add(lastName);
add(dob);
add(firstNameTF);
add(lastNameTF);
add(dobTF);
add(sbmt);
add(reset);

setSize(300,300);
setLayout(null);
```

```
setVisible(true);
}
public static void main(String[] args) {
AwtApp awt = new AwtApp();

}
}
```

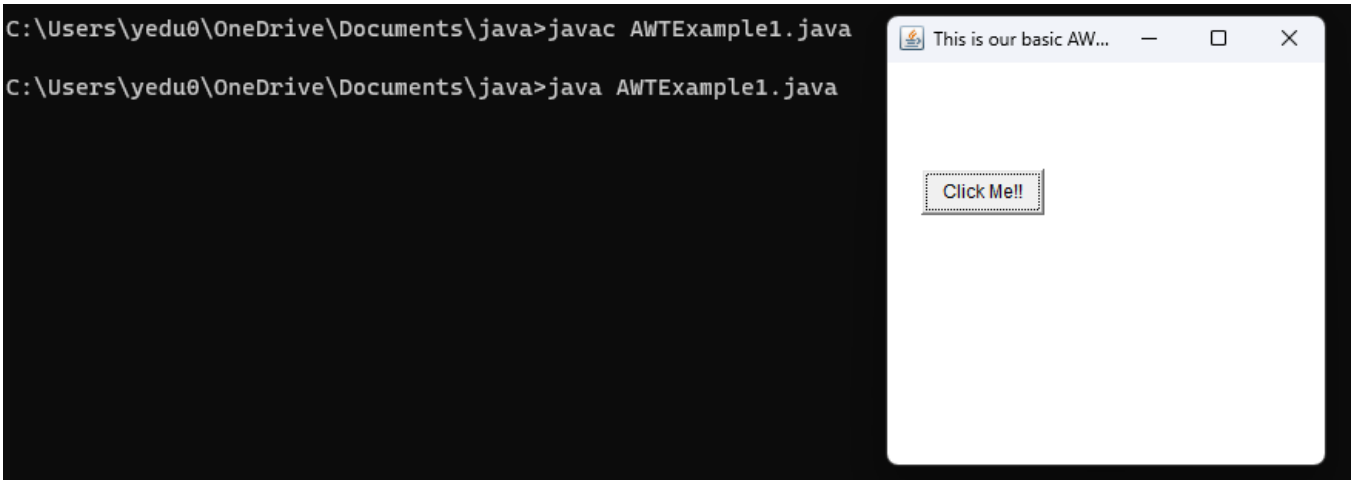


Code

```
import java.awt.*;

public class AWTExample1 extends Frame {

  AWTExample1() {

    Button b = new Button("Click Me!!");

    b.setBounds(30,100,80,30);

    add(b);

    setSize(300,300);
```

```
    setTitle("This is our basic AWT example");

    setLayout(null);

    setVisible(true);
}

public static void main(String args[]) {

AWTExample1 f = new AWTExample1();

}

}
```

Output



Conclusion:

Comment on application development using AWT Controls.

AWT (Abstract Window Toolkit) is a set of application programming interfaces (APIs) used for creating graphical user interfaces (GUIs) in Java. AWT provides a collection of classes and methods for building and managing

components such as windows, buttons, text fields, and other graphical elements. AWT Controls are the various GUI components provided by the AWT library for creating interactive user interfaces. Here's an overview of some commonly used AWT Controls:

Frame: AWT's Frame class is used to create the main application window. It serves as a container for other AWT components.

Button: The Button class represents a push button that can trigger an action when clicked.

Label: The Label class is used to display a single line of read-only text. It is often used to provide descriptions or captions for other components.

TextField: The TextField class provides a single-line input field for the user to enter text.

TextArea: The TextArea class is used to create a multi-line area for editing text.

Checkbox: The Checkbox class represents a check box that can be selected or deselected by the user.

Choice: The Choice class is used to create a drop-down list that allows the user to select one option from a list of available choices.

List: The List class is used to create a component that displays a list of items from which the user can make multiple selection.