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.

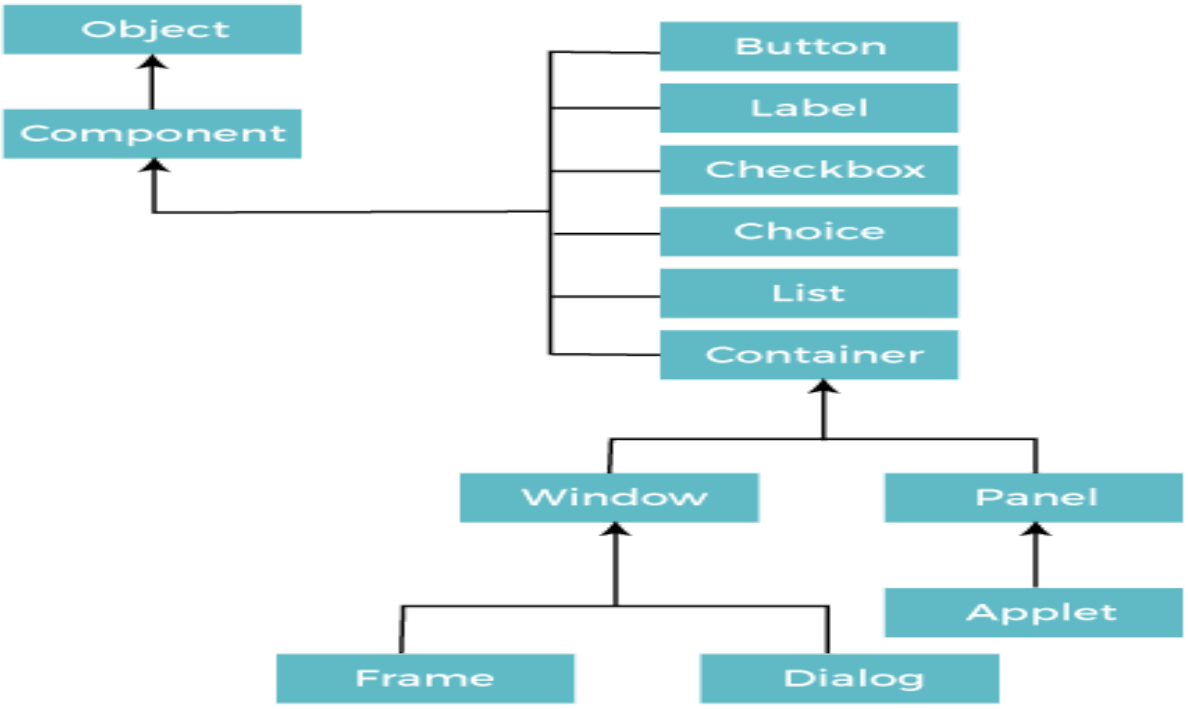
2. A basic set of GUI widgets such as buttons, text boxes, and menus1. AWT also provides Graphics and imaging tools, such as shape, color, and font classes2. AWT also avails layout managers which helps in increasing the flexibility of the window layouts2

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 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

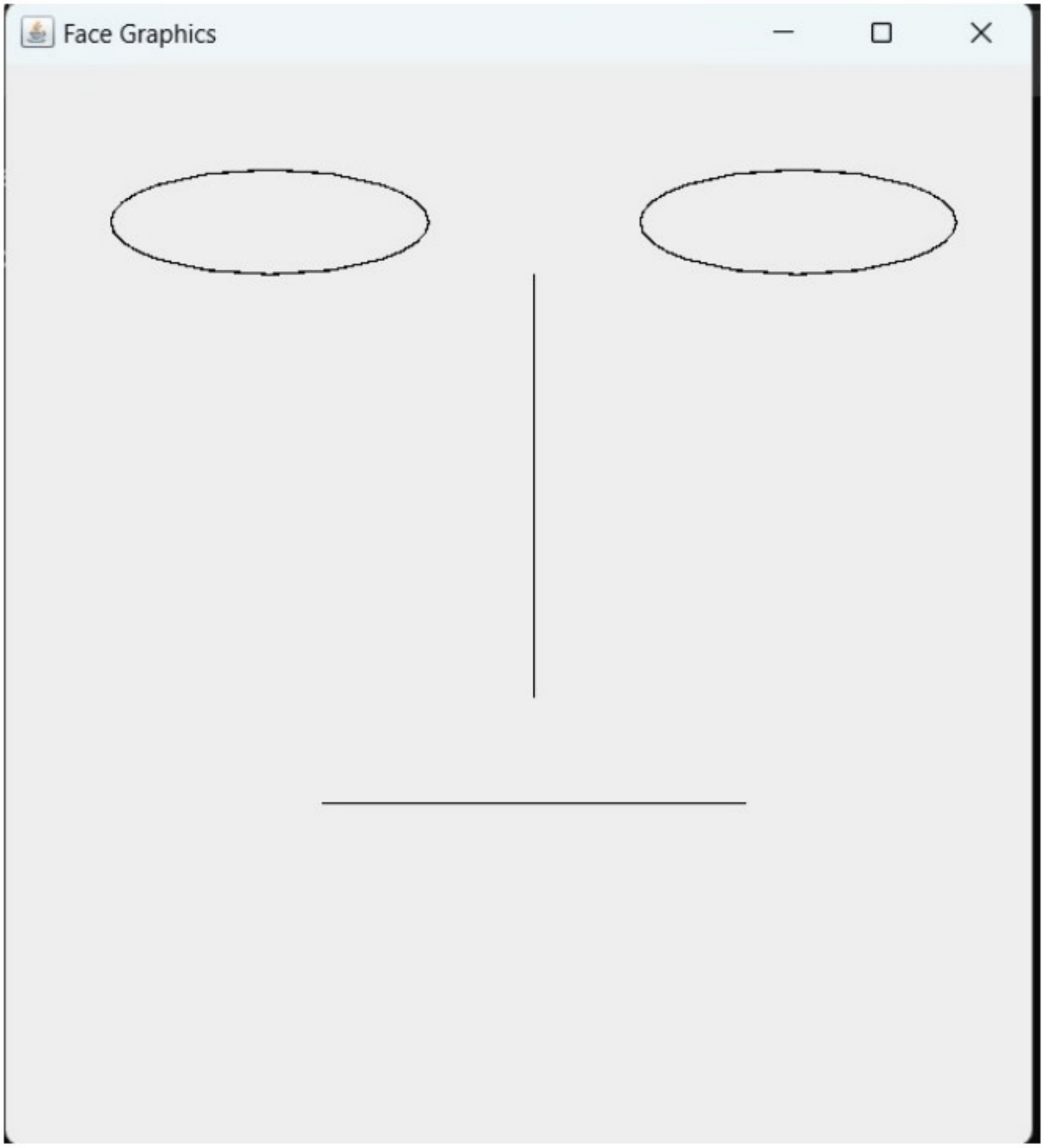


Code:

```
import javax.swing.*;
 import java.awt.*;
 class Face extends JPanel {
 @Override
protected void paintComponent(Graphics g) {
 super.paintComponent(g);
 // Drawing shapes
 g.setColor(Color.BLACK);
g.drawOval(50, 50, 150, 50);
 g.setColor(Color.BLACK);
g.drawOval(300, 50, 150, 50);
 g.setColor(Color.BLACK);
g.drawLine(250, 100, 250, 300);
 g.setColor(Color.BLACK);
 g.drawLine(150, 350, 350, 350);
```

```
} public static void main(String[] args) {
SwingUtilities.invokeLater(() -> {
JFrame frame = new JFrame("Face Graphics");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.add(new Face());
frame.setSize(500, 550);
frame.setVisible(true);
});
}
}
```

OUTPUT:



Conclusion:

Comment on application development using AWT Controls.

Application Development Using AWT Controls:

- AWT (Abstract Window Toolkit) is a Java package for creating graphical user interfaces (GUI) in Java applications.
- AWT controls, or components, include buttons, text fields, labels, checkboxes, and more.
- These controls enable developers to build platform-independent GUI applications with consistent look and feel.
- AWT applications follow an event-driven model, allowing responses to user interactions like button clicks and mouse events.
- AWT components are added to containers such as frames or panels, and their properties are set to create interactive applications.

In summary, AWT controls are fundamental for creating GUI-based Java applications, providing the foundation for user-friendly and visually appealing software.