

Programming GUI With AWT

15.1 Introduction

The **user interface** is that part of a program that interacts with the user of the program. There are two types of user interface.

- (i) Character user interface (CUI)
- (ii) Graphical user interface (GUI)

Main **advantages of building GUI applications** over CUI applications are :

- (i) GUI applications are **user-friendly**, they don't require user to have knowledge of commands (like in the DOS).
- (ii) GUI applications support **multitasking** which means that the users can perform various tasks (in other words, work with several applications) at a time. For example, One can listen to music and play a computer game. In a CUI, we need a command to complete before executing another command.
- (iii) In CUI, we can only see text, whereas in GUI, we can see images, videos and falsh.
- (iv) Examples of GUI are Windows, Mac, Ubuntu etc and DOS is an example of CUI.

In Java, there are two set of Java APIs for graphics programming.

1. AWT (Abstract Windowing Toolkit)

AWT is a concept designed for building simple GUIs in Java application. AWT is introduced in the year 1995 when java was first introduced.

AWT is platform dependent :

Java AWT calls native platform (Operating systems) subroutine for creating components such as textbox, checkbox, button etc. For example an AWT GUI having a button would have a different look and feel across platforms like Windows, Mac OS & Unix, this is because these platforms have different look and feel for their native buttons and AWT directly calls their native subroutine that creates the button. In simple, an application build on AWT would look like a windows application when it runs on Windows, but the same application would look like a Mac application when runs on Mac OS.

AWT components are considered **heavy weight** because they are being generated by underlying operating system (OS). For example if you are instantiating a text box in AWT that means you are actually asking OS to create a text box for you. Most of the AWT components have become obsolete and should be replaced by newer swing components.

2. Swing

Swing API, a much more comprehensive set of graphics libraries that enhances the AWT, was introduced as part of Java Foundation Classes (JFC) after the release of JDK 1.1.

Swing is a preferred API for window based applications because of its **platform independent** and **light-weight nature**. Swing is built upon AWT API however it provides a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

5.2 AWT Packages

AWT consists of 12 packages. But only 2 packages namely, **java.awt** and **java.awt.event** are commonly used.

1. The **java.awt** packages contains the core AWT graphics classes :
 - (i) GUI component classes (such as Button, TextField, Label etc.)
 - (ii) GUI container classes (such as Frame, Panel, Dialog etc.)
 - (iii) Layout managers (such as FlowLayout, BorderLayout, GridLayout etc.)
 - (iv) Custom graphics classes (such as Graphics, Color and Font)

2. The **java.awt.event** package supports event handling :
- (i) Event classes (such as **ActionEvent**, **MouseEvent**, **KeyEvent** and **WindowEvent**).
 - (ii) Event Listener Interfaces (such as **ActionListener**, **MouseListener**, **KeyListener** and **WindowListener**)
 - (iii) Event Listener Adapter classes (such as **MouseAdapter**, **KeyAdapter**, and **WindowAdapter**).

15.3 AWT Hierarchy

The **java.awt** package contains many classes to create GUI based applications. These classes are hierarchically arranged as shown in figure 15.1 inside the AWT package in such a manner that each successive level in the hierarchy adds certain attributes to the GUI applications.

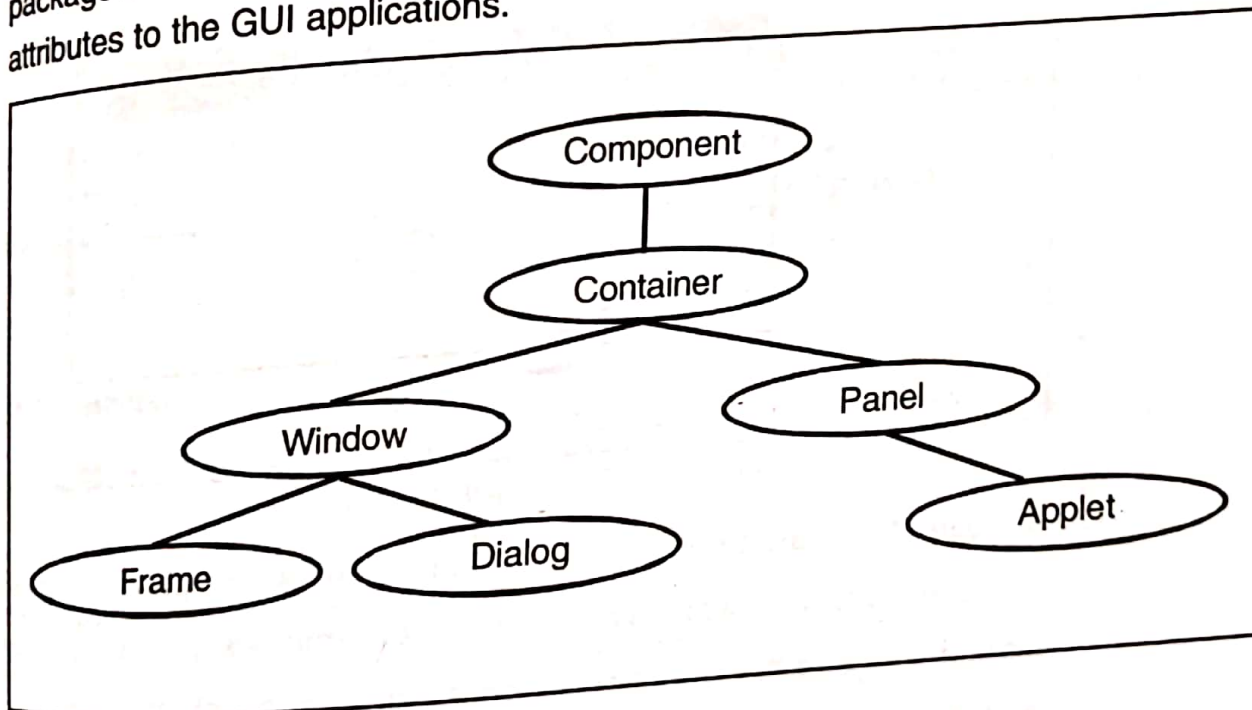


FIGURE 15.1

15.4 Components and Containers

(a) Components

A graphical user interface is built of graphical elements called **components**. Typical components include such items as buttons, scrollbars, and text fields. Components allow the user to interact with the program and provide the user with visual feedback about the state of the program. In the AWT, all user interface components are instances of class **Component** or one of its subtypes.

GUI components are also called **controls** (Microsoft ActiveX Control), **Widgets** (Google Web ToolKit) which allow users to interact with the applications through these components.

(b) Container

Containers are used to hold components in a specific layout (such as flow or grid). A container is like a screen wherein we are placing components like buttons, text fields, checkbox etc. In short a container contains and controls the layout of components. A container itself is a component thus we can add a container inside container. Figure 15.2 illustrates the concept of components and containers.

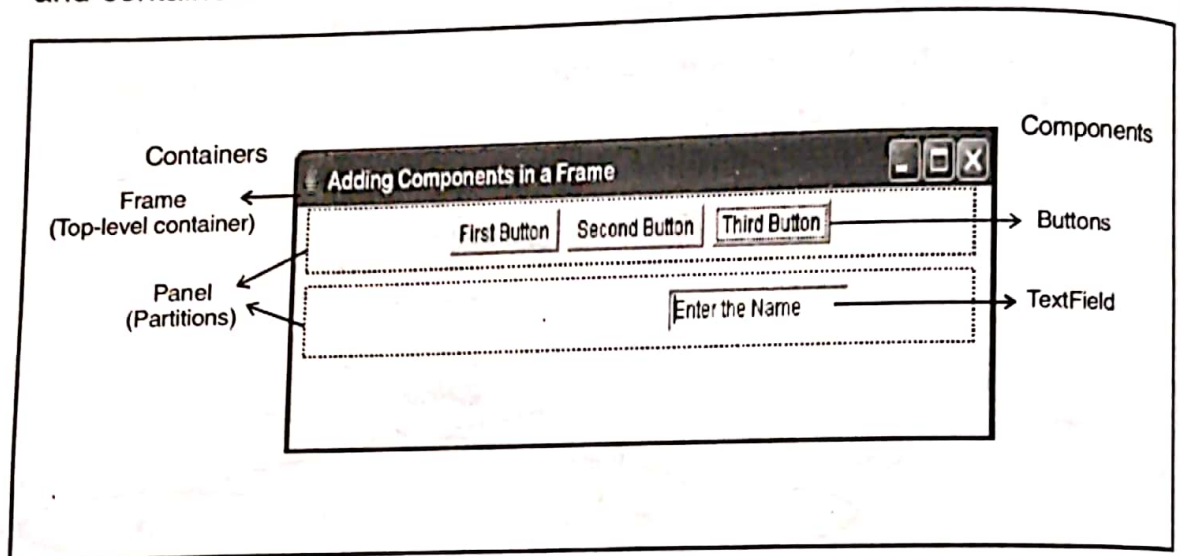


FIGURE 15.2

In figure, there are three containers; a Frame and two panels. A **frame** is the top-level container of an AWT program. A Frame has a title bar (containing an icon, a title, and the minimize/maximize/close buttons), an optional menu bar and the content display area. A **Panel** is a rectangular area used to group related GUI components in a certain layout. In the above figure, the top-level Frame contains two panels. There are four components in the above figure, a Text Field (for users to enter text) and three Buttons (for user to trigger certain programmed actions).

15.4.1 Types of Containers

The AWT provides four container classes. They are class Window and its two

subtypes, namely, class Frame and class Dialog, as well as the Panel class. In addition to the containers provided by the AWT, the Applet class is a container, it is subtype of the Panel class and can therefore hold components.

(a) Frame

A Frame provides the "main window" for the GUI application, which has a title bar (containing an icon, a title, the minimize, maximize/restore-down and close buttons), an optional menu bar, and the content display area as shown in figure 15.3.

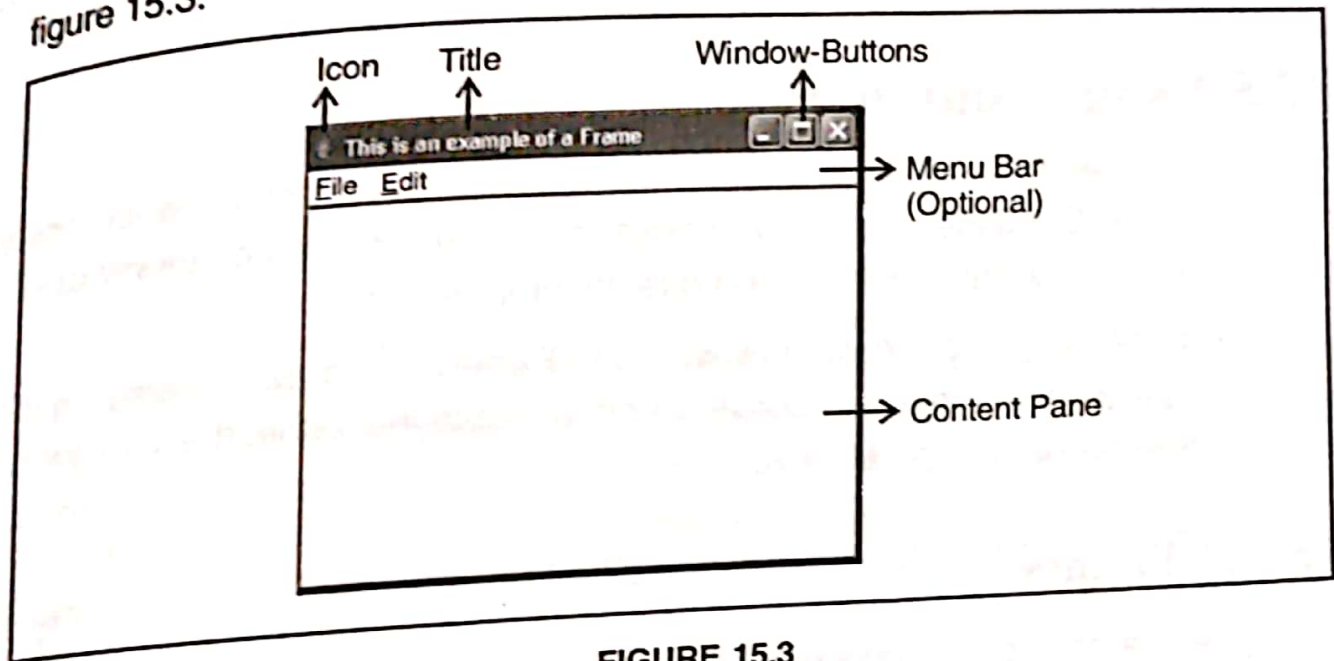


FIGURE 15.3

(b) Dialog

An AWT Dialog is a "pop-up window" used for interacting with the users. A Dialog has a title-bar (containing an icon, a title and a close button) and a content display area, as shown in figure 15.4.

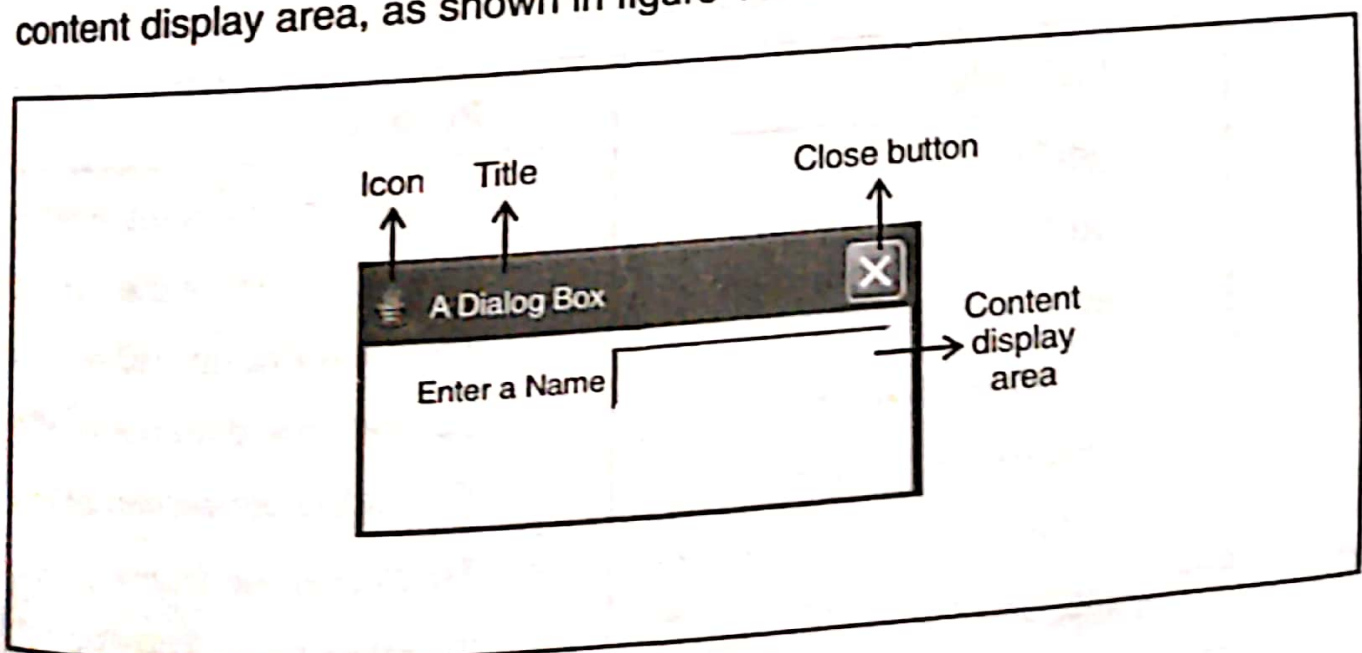


FIGURE 15.4

(c) Window

A window in AWT is a container represents a top-level window that doesn't have a title bar (so it doesn't have close, minimize, maximize buttons), menu bar. This sits on the desktop directly and has a BorderLayout by default.

(d) Panel

It is a sub class of the Container class. It is a border less window that doesn't contain any title bar or menubar. Usually added to another container like the Frame. The main purpose of a panel is to group components. It comes with a default FlowLayout.