Chapter 9: GUI with AWT and Swing

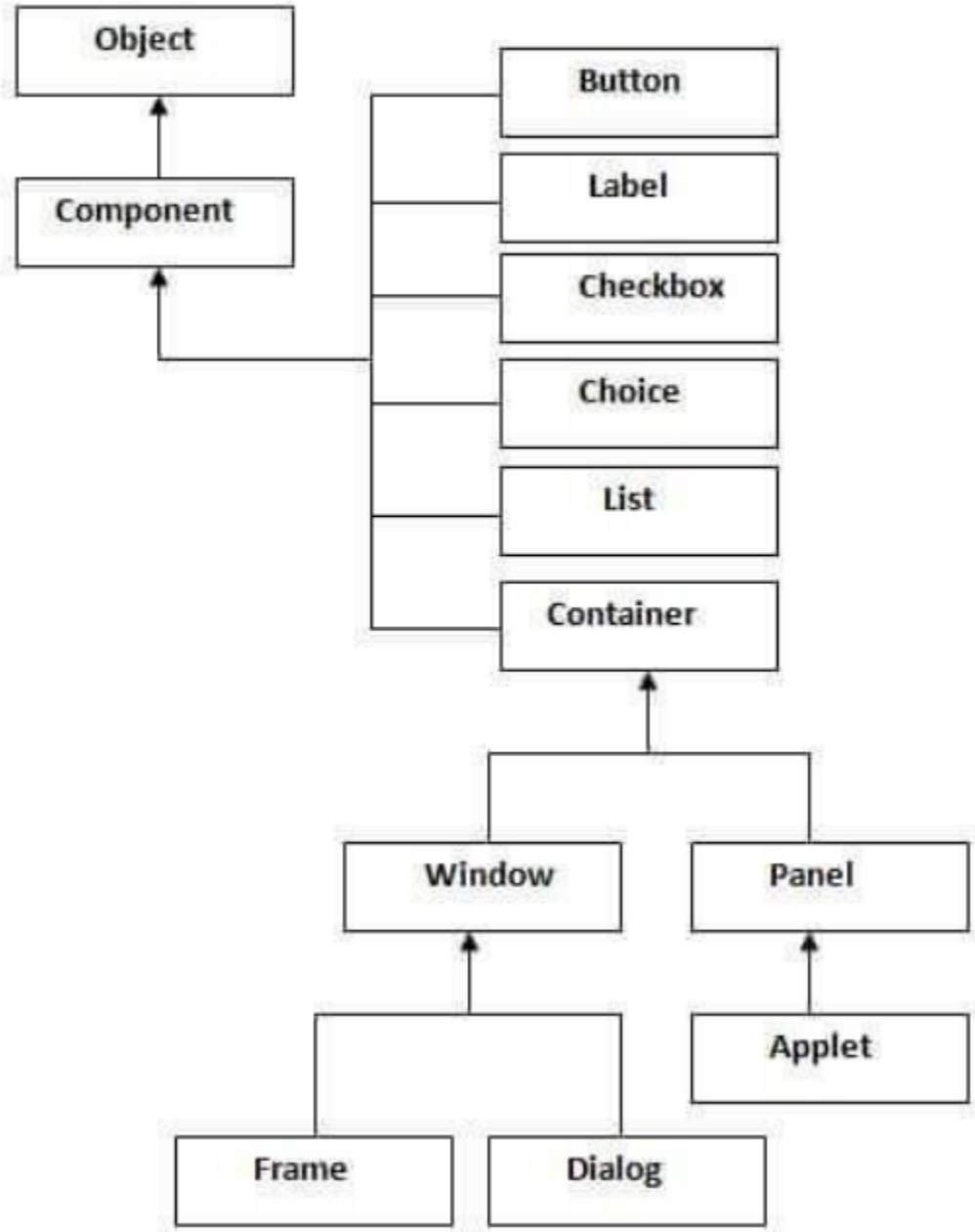
Graphical User Interface(GUI):

- interface that allows users to operate an application and perform certain tasks .
- GUI allows the user to communicate with the application.
- GUI provides results to the end user in the response to events that are raised.
- The activities like clicking a button, closing a window, opening window etc. can be considered as events.
- It consists of components like buttons, labels,drop down lists etc.
- AWT and Swing are popular ways to develop GUI based applications.

Abstract Window Toolkit(AWT):

- Platform dependent API used to develop GUI or windows-based applications in Java.
- For developing an AWT based GUI, the programmer has to import the javax.awt package .
- Some AWT components are buttons, scrollbars, text fields, lists, dialogs, and panels.
- It is heavy weight as its components uses the resources of the operating system.
- It consists of a large number of classes and methods, which are used for creating and managing GUI.
- AWT components are platform dependent.i.e. The look and feel of the applications created using AWT depends on the operating system.
- The java.awt package provides various classes for awt api like Label, Textfield,RadioButton, List etc.

Hierarchy of AWT:



Component class:

- It is an abstract class and lies at the top of AWT hierarchy.
- The component object consists of the information regarding foreground, background color,currently selected text color etc.
- It handles different mouse and keyboard events of GUI applications.
- The elements like buttons, scrollbar, labels etc. are called components.
- For keeping different components on a particular place, we need to add the components to a container.

Container class:

- It is the component in AWT which contains other components like buttons, textfields, labels etc.
- It is the container which contains other components.
- Other classes like Panel, Dialog ,Frame are derived from the Container class.
- It controls the layout of components.

Window class:

- The window is a subclass of container class which has no borders and menu bars.
- For creating a window, we must need a frame, dialog, or another window.

Panel:

- It is the subclass of container class which donot have title bar and menu bars.
- It have other components like button, textfields etc.

Applet:

- It is the subclass of panel which is used to develop programs that run in web browsers.

Frame:

- It is the subclass of the Window class which has a title bar and menu bars.
- Also, it have other components like button, textfields etc.
- It is most widely used for developing AWT applications.

Dialog:

- It is the subclass of Window class .
- It consists of the title bar.

Methods of Component class:

Method	Description
public void add(Component c)	inserts a component on this component.
public void setSize(int width,int height)	sets the size (width and height) of the component.
public void setLayout(LayoutManager m)	defines the layout manager for the component.
public void setVisible(boolean status)	changes the visibility of the component, by default false.

Frame:

- It is the AWT container used to display a collection of graphical AWT components.

- The size of the frame contains the area designated for the border.

Program Example1:Illustrating Frame

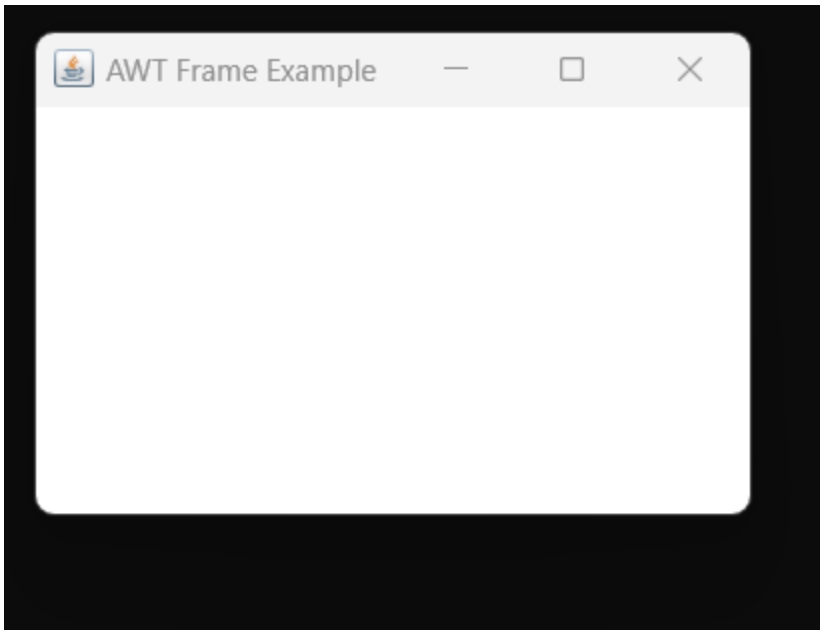
```
import java.awt.*;

public class Example1Frame {
  public static void main(String[] args) {
      // Create a Frame
      Frame frame = new Frame("AWT Frame Example");

      // Set Frame size
      frame.setSize(300, 200);


      // Make the Frame visible
      frame.setVisible(true);
  }
}
```

Output:



Example illustrating Panel:

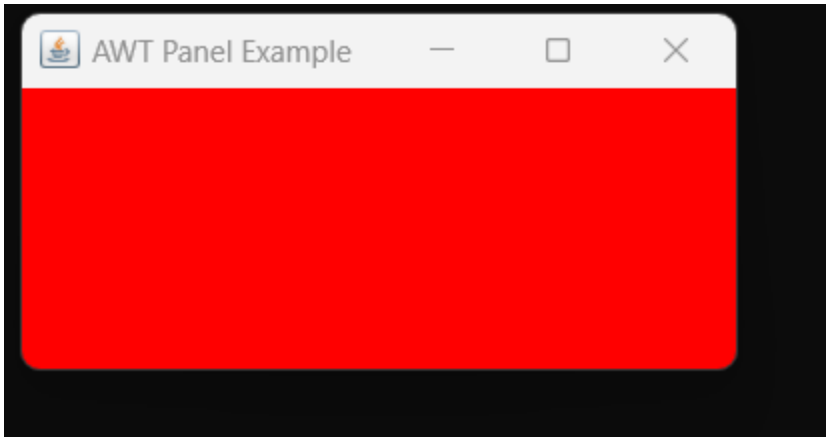
```
import java.awt.*;

public class Example2Frame {
  public static void main(String[] args) {
      // Create a Frame
      Frame frame = new Frame("AWT Panel Example");

      // Set Frame size
      frame.setSize(300, 150);
        Panel panel = new Panel();
         panel.setBackground(Color.red);
      // Add Panel to the Frame
      frame.add(panel);

      // Make the Frame visible
      frame.setVisible(true);
  }
}
```

Output:



AWT Components:

1. **Label:** Label refers to the non-editable text component for displaying the information.

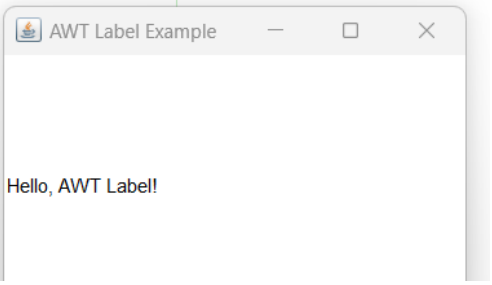
Example:

```java
import java.awt.*;

public class AWTLabelExample {
    public static void main(String[] args) {
        Frame f = new Frame("AWT Label Example");

        Label l = new Label("Hello, AWT Label!");

        f.add(l);
        f.setSize(300, 200);
        f.setVisible(true);
    }
}
```



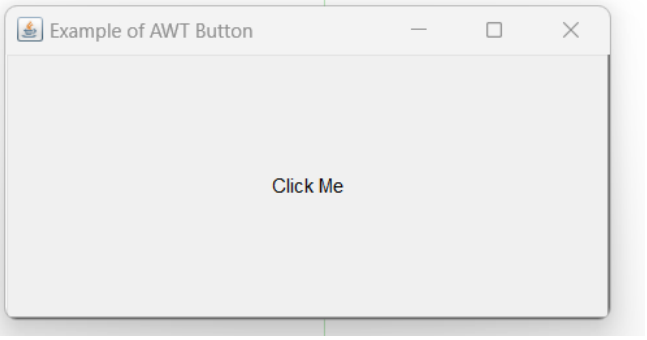
2. **AWT Button:** It refers to the component that is clicked by the users to trigger actions.

Example:

```java
import java.awt.*;

public class AWTButtonExample {
   public static void main(String[] args) {
        Frame frame = new Frame("Example of AWT Button");

        Button button = new Button("Click Me");
        frame.add(button);
        frame.setSize(300, 200);
        frame.setVisible(true);
   }
}
```



3. AWT Textfield:

- Textfield is used for allowing the users to input the single-line text.

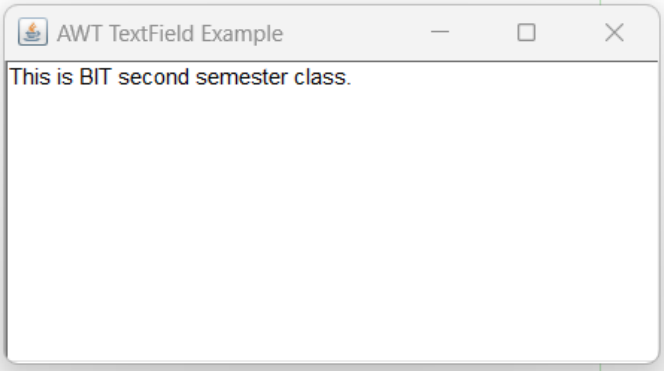
Example:

```java
import java.awt.*;

public class AWTTextFieldExample {
   public static void main(String[] args) {
        Frame f = new Frame("AWT TextField Example");

        TextField t = new TextField("Type here");

        f.add(t);
        f.setSize(300, 200);
        f.setVisible(true);
   }
}
```



4. AWT TextArea:

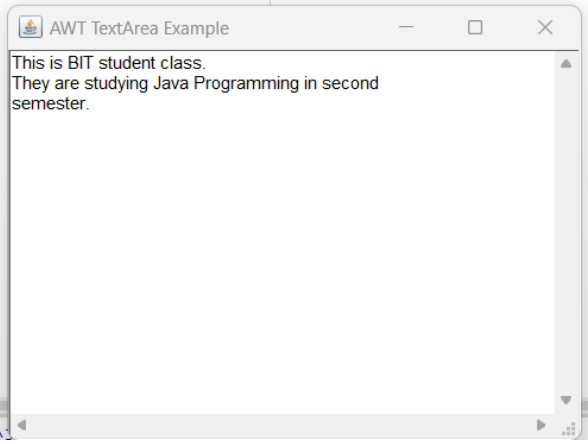
- It allows users to input multi-line text.

```java
import java.awt.*;

public class AWTTextAreaExample {
    public static void main(String[] args) {
        Frame f = new Frame("AWT TextArea Example");

        TextArea ta = new TextArea("Type here");

        f.add(ta);
        f.setSize(400, 300);
        f.setVisible(true);
    }
}
```




```
AreaExample.java" (in directory: C:\Users\ASUS\Desktop\
```

5. AWT Checkbox:

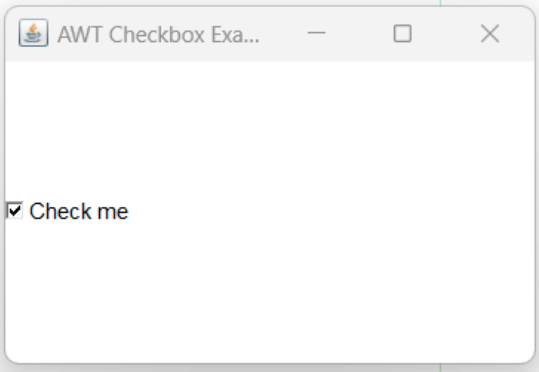
- It allows for binary choice control.

```java
import java.awt.*;

public class AWTCheckboxExample {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT Checkbox Example");

        Checkbox checkbox = new Checkbox("Check me");

        frame.add(checkbox);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```



6. AWT RadioButton:

- It allows one to choose from multiple choices.

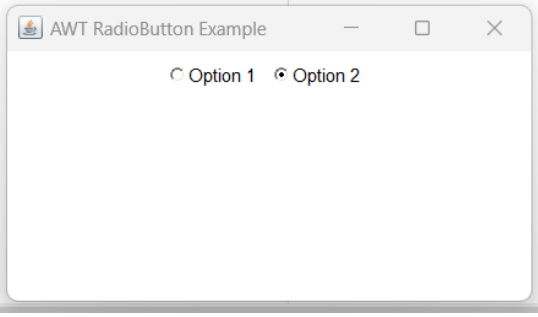
Example:

```
import java.awt.*;

public class AWTRadioButtonExample {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT RadioButton Example");

        CheckboxGroup group = new CheckboxGroup();
        Checkbox radio1 = new Checkbox("Option 1", group, false);
        Checkbox radio2 = new Checkbox("Option 2", group, true);

        frame.add(radio1);
        frame.add(radio2);
        frame.setLayout(new FlowLayout());
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```



7. AWT Choice:

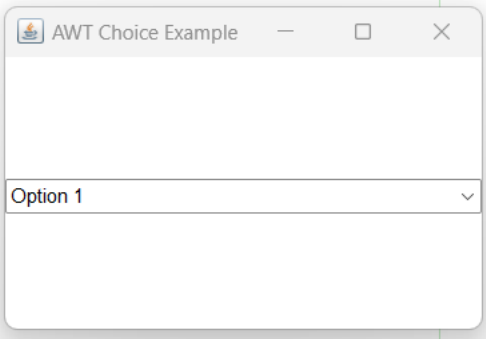
- It allows to make dropdown list of the selectable items.

```
import java.awt.*;

public class AWTChoiceExample {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT Choice Example");

        Choice choice = new Choice();
        choice.add("Option 1");
        choice.add("Option 2");
        choice.add("Option 3");

        frame.add(choice);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```



8. AWT List:

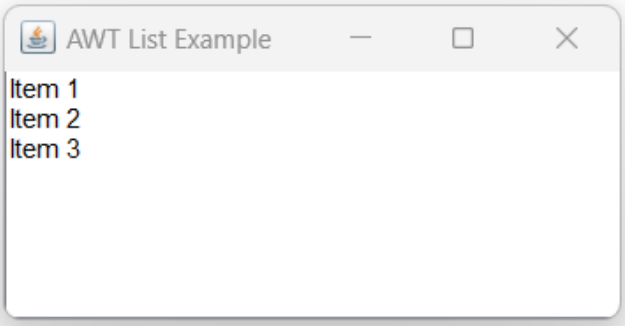
- It allows us to display a scrollable list of items.

```java
import java.awt.*;

public class AWTListExample {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT List Example");

        List list = new List(3, false);
        list.add("Item 1");
        list.add("Item 2");
        list.add("Item 3");

        frame.add(list);
        frame.setSize(200, 150);
        frame.setVisible(true);
    }
}
```



9. **AWT Scrollbar:**

- It allows you to scroll through the content.

Example:

```java
import java.awt.*;

public class AWTScrollbarExample {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT Scrollbar Example");

        Scrollbar scrollbar = new Scrollbar();

        frame.add(scrollbar);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```