**1. How can we add GUI to a Java program?**

To add a GUI to a Java program, you can use GUI libraries that Java provides, such as **AWT (Abstract Window Toolkit)** and **Swing**. These libraries allow you to create graphical components like windows, buttons, text boxes, and labels to build interactive applications.

**General Steps to Add GUI:**

1. **Import the GUI Library**:
   * Use import javax.swing.\*; for Swing or import java.awt.\*; for AWT to import the necessary GUI classes.
2. **Create a Frame**:
   * You can create a window by using the JFrame class.
3. **Add Components**:
   * Add components like buttons (JButton), labels (JLabel), text fields (JTextField), etc., to the frame.
4. **Set Layout**:
   * Use a layout manager (such as BorderLayout, FlowLayout) to arrange components in the frame.
5. **Display the Frame**:
   * Call setVisible(true) on the frame to make it visible.

**Example**

Here is a simple Java program using Swing to create a GUI:

*import javax.swing.\*;*

*public class SimpleGUI {*

*public static void main(String[] args) {*

*// Create a JFrame window*

*JFrame frame = new JFrame("My GUI Program");*

*frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);*

*frame.setSize(300, 200);*

*// Create a button and add it to the frame*

*JButton button = new JButton("Click Me");*

*frame.getContentPane().add(button);*

*// Make the frame visible*

*frame.setVisible(true);*

*}*

*}*

In the above code, a simple window with a button is created. The user can interact with the button.

**2. What is AWT (Abstract Windowing Toolkit)?**

**AWT (Abstract Window Toolkit)** is one of the earliest graphical user interface (GUI) toolkits provided by Java. It is a part of the Java core libraries and allows developers to create GUI components like windows, buttons, labels, and text boxes.

**Features:**

* **Platform Dependent**: AWT uses the native platform's GUI components, meaning its appearance can vary depending on the operating system. It is **platform-dependent**.
* **Basic Components**: AWT provides basic GUI components such as Button, Label, TextField, etc.
* **Event Handling**: AWT handles events (such as user clicks or key presses) using listeners like ActionListener, MouseListener, etc.

AWT was the first GUI library in Java, but it had some limitations, leading to the development of **Swing**, a more advanced GUI toolkit.

**3. What is Swing?**

**Swing** is a more advanced GUI toolkit that was introduced after AWT, extending its capabilities and providing a richer set of GUI components.

**Features:**

1. **Lightweight Components**:
   * Swing components are **lightweight**, meaning they do not depend on the native operating system's GUI components. This makes Swing components appear consistent across all platforms.
2. **Rich Set of Components**:
   * Swing provides a more diverse set of components compared to AWT, such as JButton (button), JLabel (label), JTextField (text field), etc. All Swing components typically start with the letter J (e.g., JFrame, JPanel).
3. **MVC Architecture**:
   * Swing follows the **Model-View-Controller (MVC)** architecture, which separates the display of a component from its behavior, allowing for greater flexibility and maintainability.
4. **Look and Feel**:
   * Swing allows developers to change the look and feel of the components, making the application more attractive and customizable.

**Example: Swing Window**

Here is an example of creating a simple GUI window using Swing:

*import javax.swing.\*;*

*public class SwingExample {*

*public static void main(String[] args) {*

*// Create a JFrame window*

*JFrame frame = new JFrame("Swing Example");*

*frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);*

*frame.setSize(400, 300);*

*// Create a label and add it to the frame*

*JLabel label = new JLabel("Hello, Swing!");*

*frame.getContentPane().add(label);*

*// Make the frame visible*

*frame.setVisible(true);*

*}*

*}*

In this example, we create a window using the JFrame class and add a JLabel to display text in the window.