**I. Comparison of Top-Level Containers**

| **Aspect** | **AWT** | **Swing** |
| --- | --- | --- |
| **Main Window** | **Frame**: A heavyweight component | **JFrame**: A lightweight container |
| **Additional Window** | **Window**: A top-level container similar to Frame but without a title bar, border, or menu bar. Used for creating additional windows or dialogs. | **JDialog**: Similar to AWT's Window but provides more features and flexibility, including support for Swing components and dialog-specific functionality like modal behavior. |
| **Web-based**  **Application** | **Applet**: A container for web applications. Runs in a browser or applet viewer with security restrictions. | **JApplet**: The Swing equivalent of Applet, allowing the inclusion of Swing components in a web-based environment. It is lightweight. |

**II. Comparison of Component Class Names**

| **Function** | **AWT Class** | **Swing Class** |
| --- | --- | --- |
| Button | Button | JButton |
| Checkbox | Checkbox | JCheckBox |
| Choice (Dropdown) | Choice | JComboBox |
| Label | Label | JLabel |
| List | List | JList |
| Scroll Pane | ScrollPane | JScrollPane |
| Scroll Bar | Scrollbar | JScrollBar |
| Text Area | TextArea | JTextArea |
| Text Field | TextField | JTextField |
| Panel | Panel | JPanel |
| Frame | Frame | JFrame |
| Window (Dialog) | Window | JDialog |
| Menu | Menu | JMenu |
| Menu Bar | MenuBar | JMenuBar |
| Menu Item | MenuItem | JMenuItem |
| Popup Menu | PopupMenu | JPopupMenu |
| Text Component | TextComponent | JTextComponent |
| Dialog | Dialog | JDialog |

**III. Comparison of Event-Handling**

| **Aspect** | **AWT** | **Swing** |
| --- | --- | --- |
| **Model** | Delegation event model: Event handling is done by delegation. Components have corresponding listener interfaces (e.g., **ActionListener**, **MouseListener**). | Listener model: Similar to AWT, Swing uses listener interfaces for event handling, providing a rich set of listeners for various events. |
| **Anonymous Inner Classes** | Commonly used for implementing event listeners, which can make code more compact but may affect readability and maintainability for larger applications. | Commonly used, but Swing also supports separate listener classes and lambda expressions (introduced in Java 8) for better readability and maintainability. |
| **Concurrency** | Typically uses a single event dispatch thread (EDT) to handle all UI events. Long-running tasks in the EDT can cause UI unresponsiveness. | Ensures UI updates and event handling are performed on the EDT. Utilities like **SwingUtilities.invokeLater()** and **SwingWorker** help manage concurrency and prevent UI responsiveness issues. |
| **Custom Events** | Custom event creation is less common. | Allows creation of custom events and event listeners, useful for application-specific event handling. |
| **Look and Feel** | Limited customization options. | Pluggable look-and-feel architecture allows customization of appearance and behavior independently of the platform, including event handling related to user interactions with GUI components. |