Introduction to AWT package

- Abstract Window Toolkit in Java, , enables the programmers to create GUI-based applications.
- AWT provides support for both standard & applet windows.
- It contains a number of classes that help to implement common window-based tasks such as :Manipulating windows, adding scroll bars, buttons, list items, text boxes etc.
- All the classes are contained in the **java.awt** package.

Following fig shows how, the corresponding classes are hierarchically arranged in the awt package:

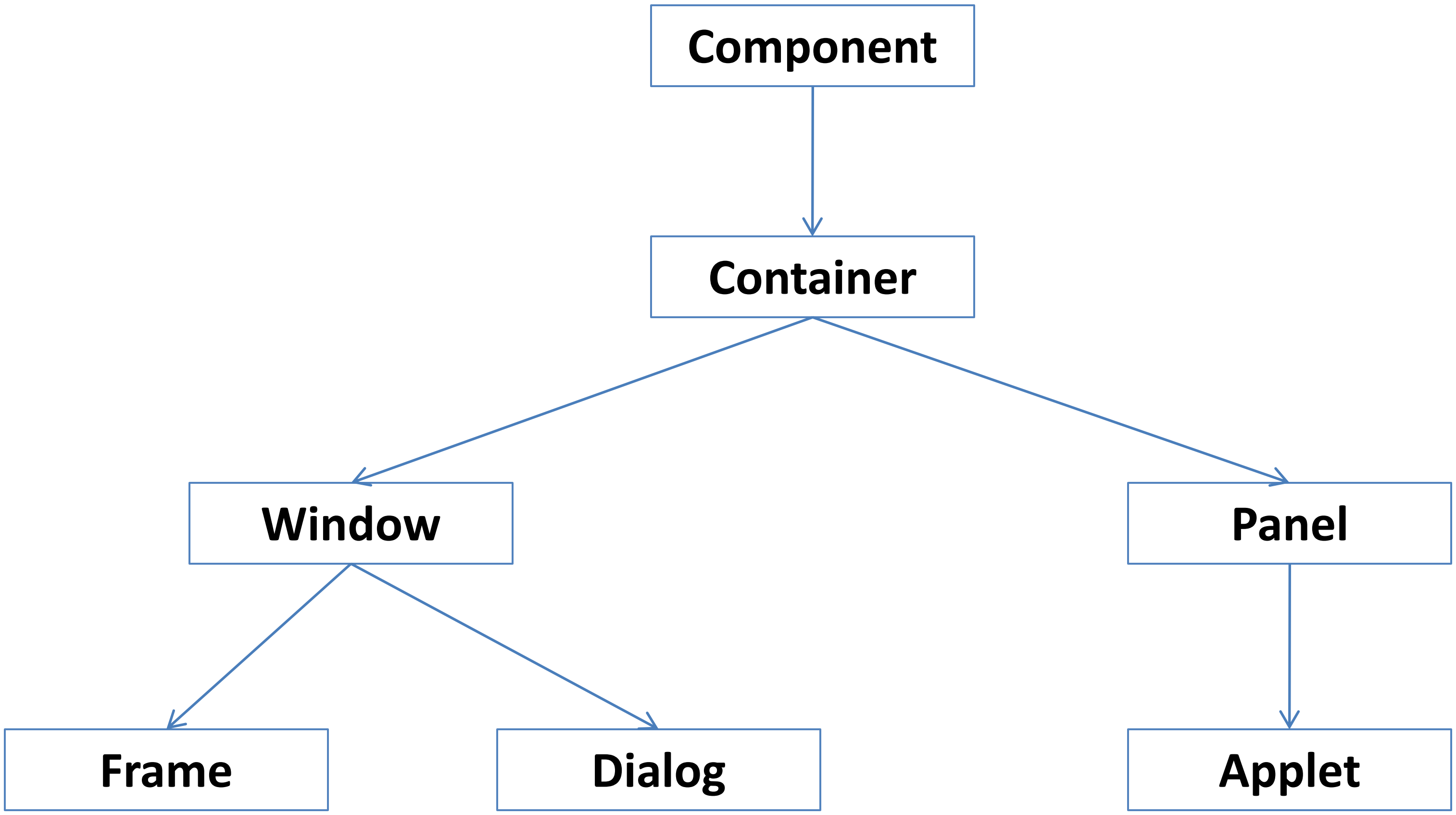


Fig.AWT Hierarchy

❖ Component:

- It is the super class to all the other classes from which various GUI elements are realized.
- Primarily responsible for effecting the display of a graphic object on the screen.
- Also handles various keyboard and mouse events of the GUI application.

❖ Container:

- Container object contains the other awt components.
- It manages the layout and placement of various awt components within the container.
- A container object can contain another container objects as well, thus allowing nesting of containers.

❖ Window:

- The window object realizes a top level window but without any border or menu bar.
- It just specifies the layout of the window.
- A typical window that you want to create in your applications is not normally derived from the window class but from its subclass: i.e. frame.

❖ Panel:

- The super class of applet, panel represents a window space on which the application’s output is displayed.
- It is just like a normal window having no border, title bar, menu bar etc.
- A panel can contain within itself, other panels as well.

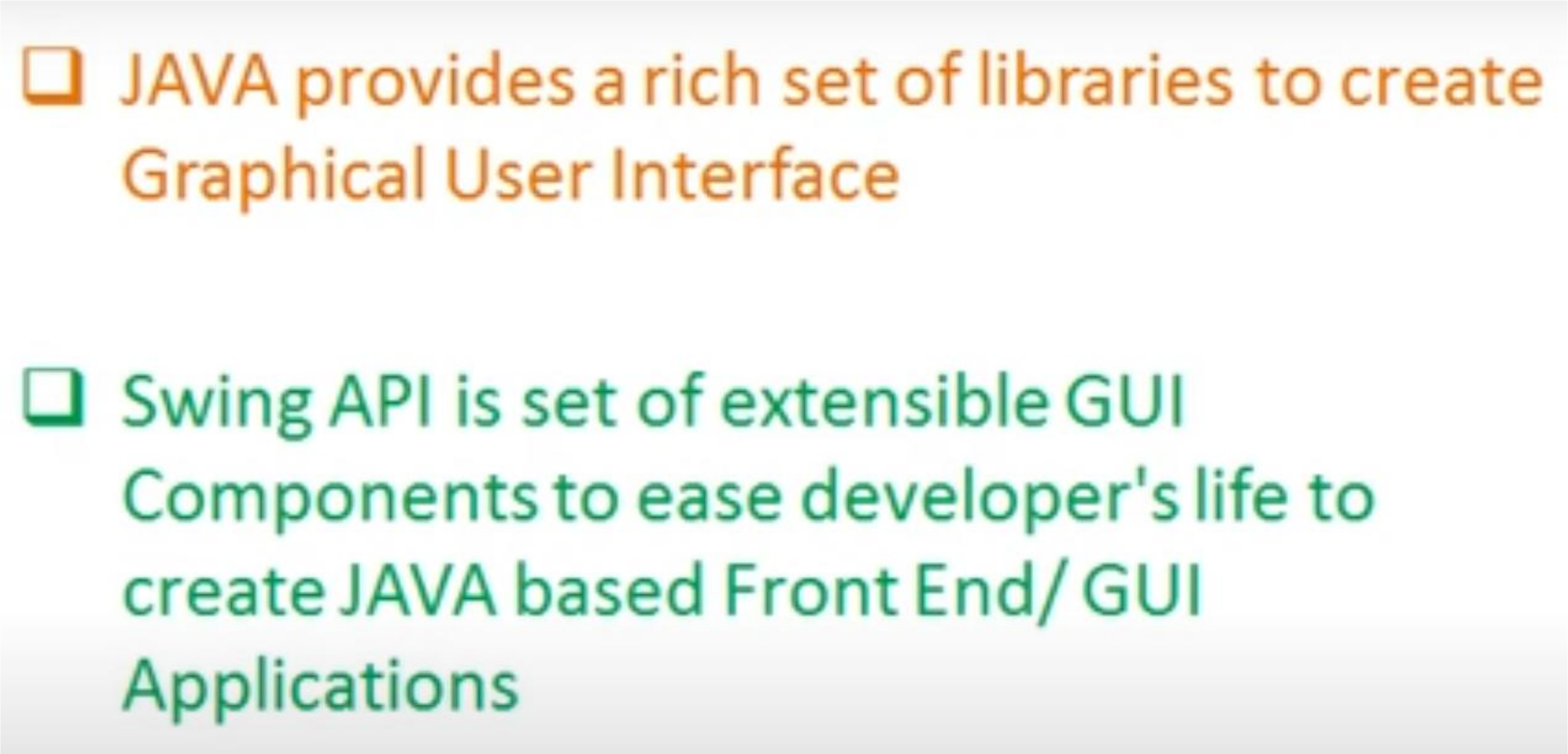
❖ Frame:

- The frame object realizes a top level window, complete with border and menu bar.
- It supports common window related events such as, close, open, activate, deactivate etc.

Almost all the programs that we created while discussing applets and graphics programming used one or more classes of the awt package.

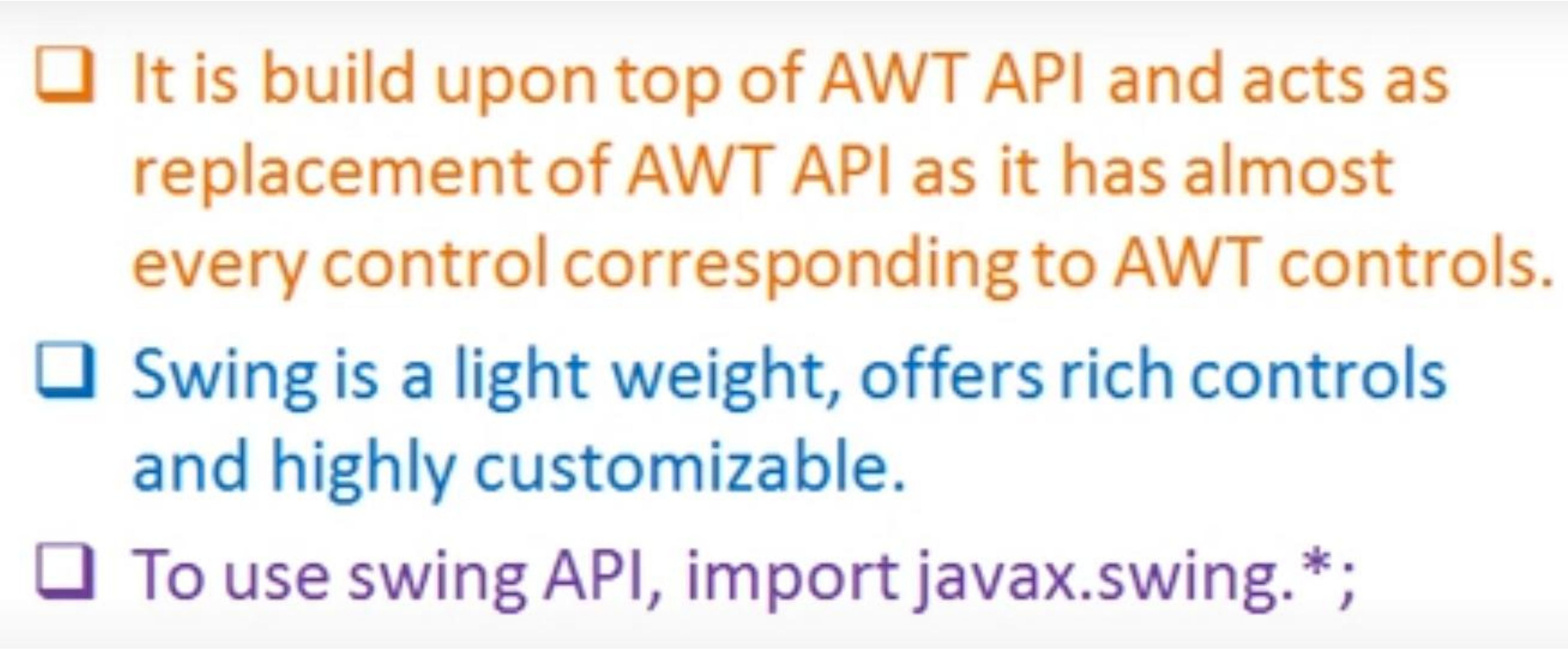
Introduction to Swings

- Similar to awt, swing is also a GUI toolkit that facilitates the creation of highly interactive GUI applications.
- Swing is more flexible & robust when it comes to implementing graphical components.
- One major difference: swing will always generate similar type of out put irrespective of the underlying platform.
- Awt on the other hand, is more dependent on the underlying operating system, which may vary the output from one platform to another.



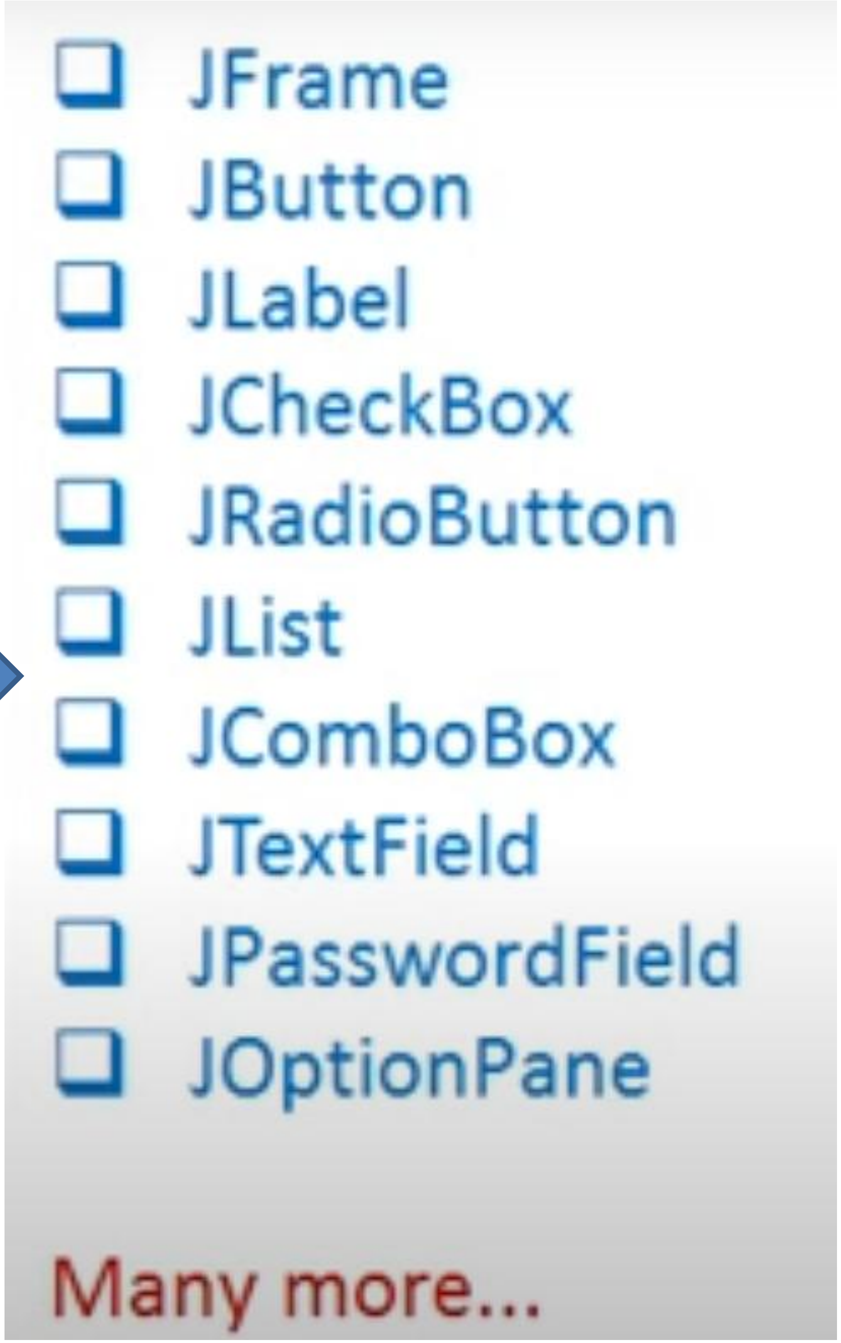
□ JAVA provides a rich set of libraries to create Graphical User Interface

□ Swing API is set of extensible GUI Components to ease developer's life to create JAVA based Front End/ GUI Applications



- □ It is build upon top of AWT API and acts as replacement of AWT API as it has almost every control corresponding to AWT controls.
- □ Swing is a light weight, offers rich controls and highly customizable.
- □ To use swing API, import javax.swing.*;

Swing Components



- ☐ JFrame
- ☐ JButton
- ☐ JLabel
- ☐ JCheckBox
- ☐ JRadioButton
- ☐ JList
- ☐ JComboBox
- ☐ JTextField
- ☐ JPasswordField
- ☐ JOptionPane

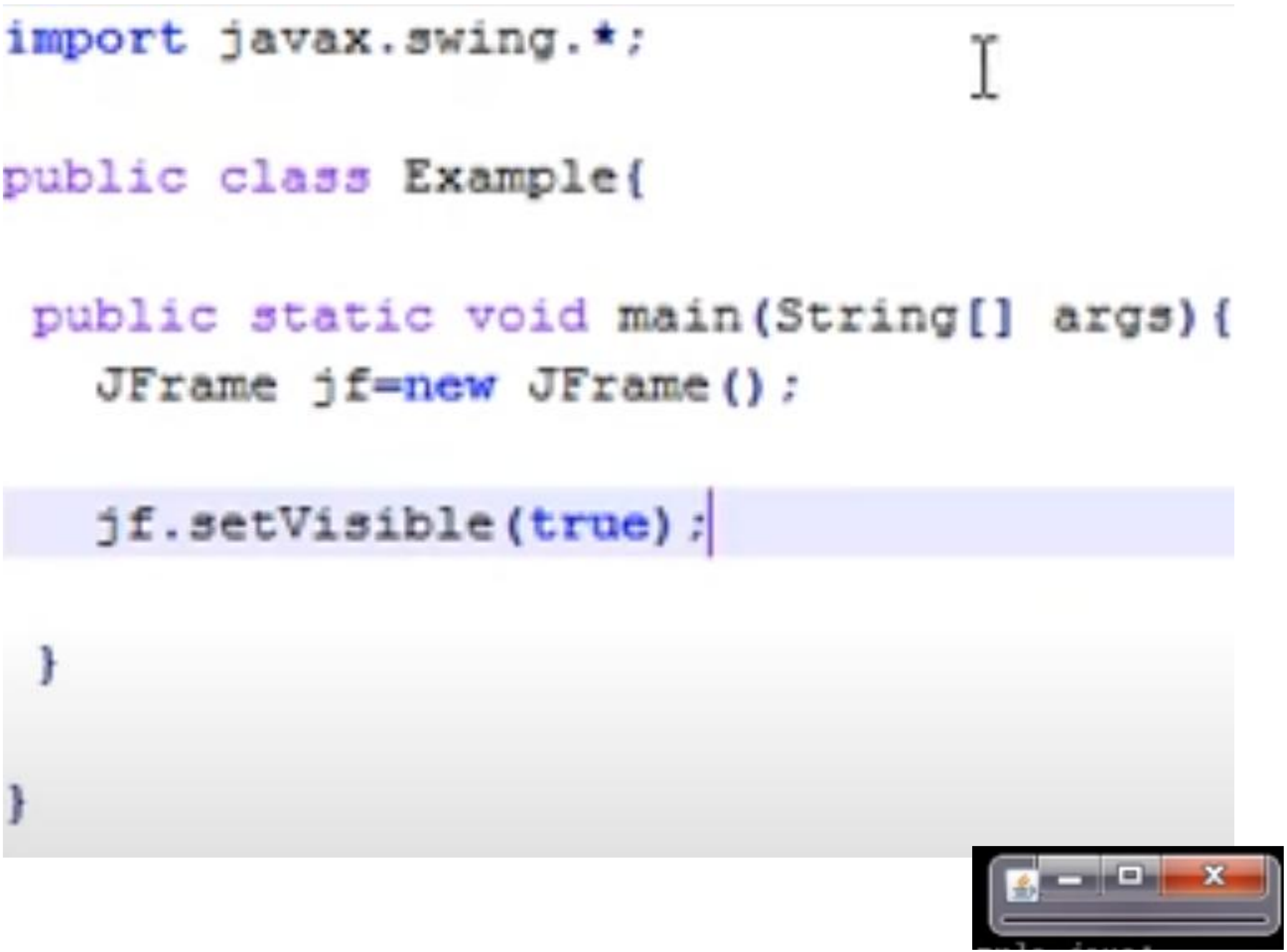
Many more...

- Swing is more graphically rich than awt.
- They provide some entirely new graphical components like tabbed window and tree structure.
- They have also enhanced some conventional awt components like buttons with both images & texts.

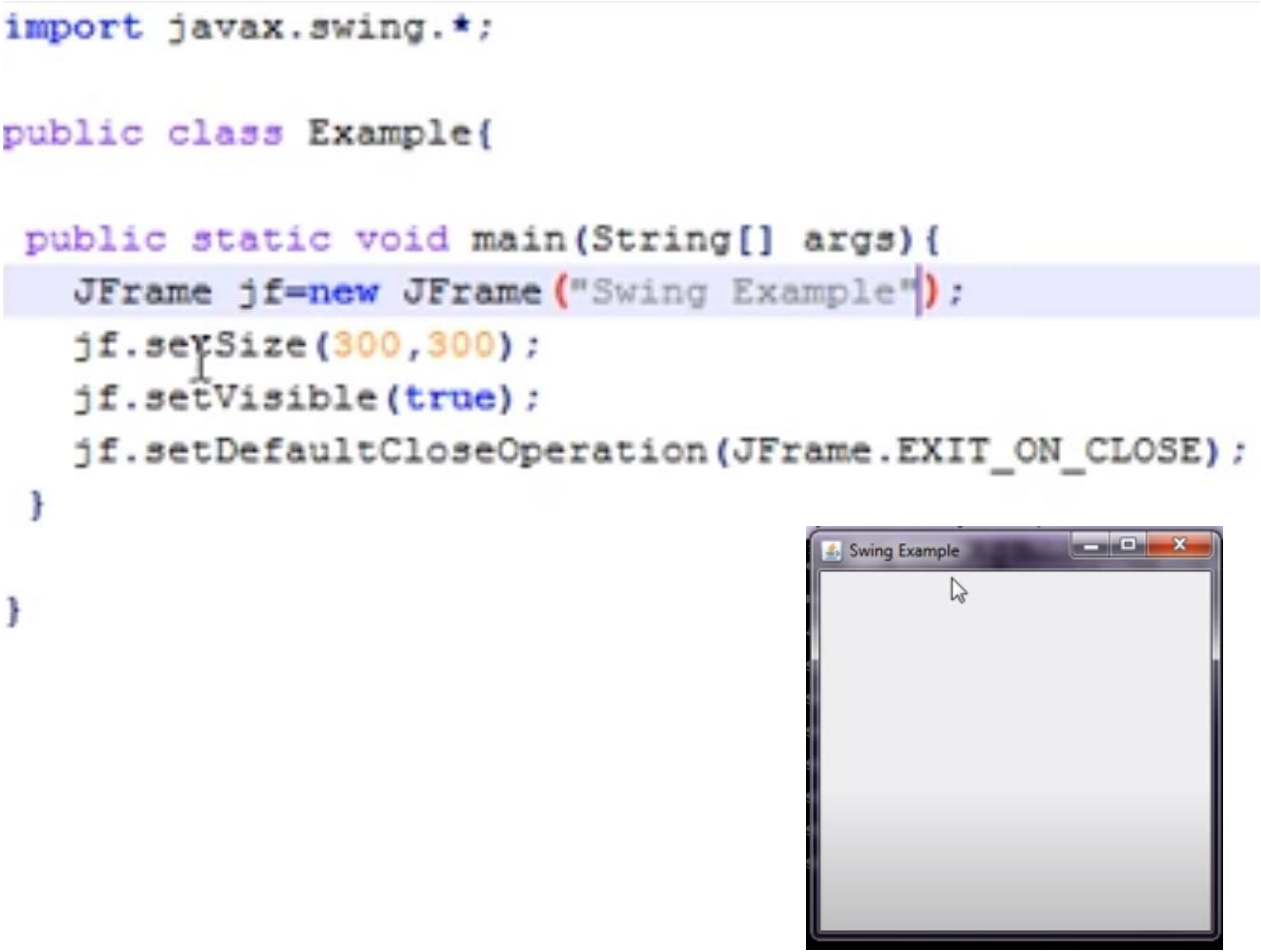
Some of the key swing classes are:

- JApplet: an extension of applet class, it is the swing's version of applet.
- JFrame: an extension of **java.awt.Frame** class, it is swing's version of frame.
- JButton: Helps to realize a push button in swing.
- JTabbedPane: Helps to realize a tabbed pane in swing.
- Jtree: Helps to realize a hierarchical tree structure in swing.
- JComboBox: Helps to realize a combo box in swing.

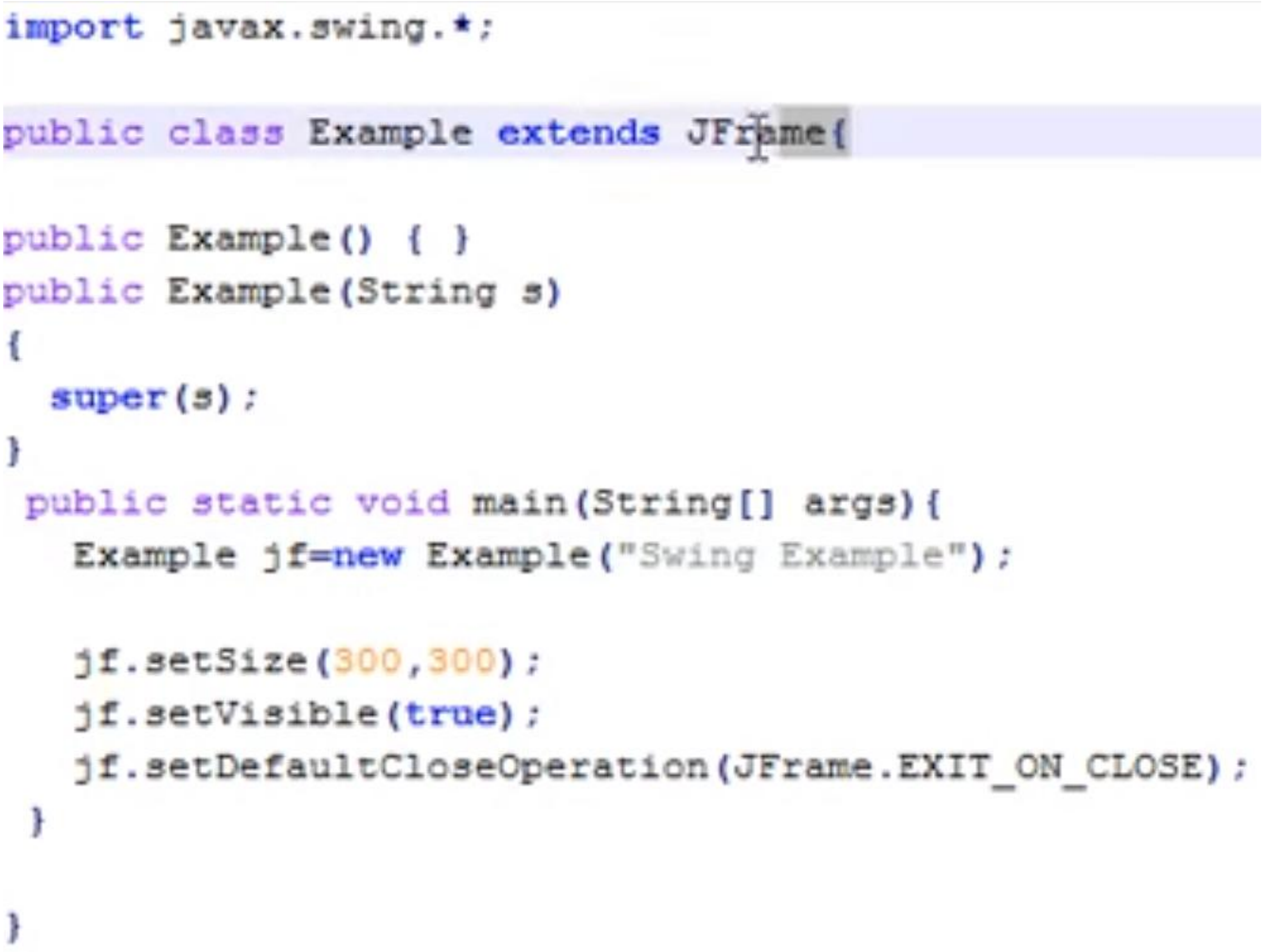
AWT	Swing
AWT stands for Abstract windows toolkit.	Swing is also called as JFC's (Java Foundation classes).
AWT components are called Heavyweight component.	Swings are called light weight component because swing components sits on the top of AWT components and do the work.
AWT components require java.awt package.	Swing components require javax.swing package.
AWT components are platform dependent.	Swing components are made in purely java and they are platform independent.
This feature is not supported in AWT.	We can have different look and feel in Swing.
These feature is not available in AWT.	Swing has many advanced features like JTabel, Jtabbed pane which is not available in AWT.
AWT is a thin layer of code on top of the OS.	Swing is much larger. Swing also has very much richer functionality.
Using AWT, you have to implement a lot of things yourself.	Swing has them built in.



```
import javax.swing.*;

public class Example{

 public static void main(String[] args){
   JFrame jf=new JFrame();

   jf.setVisible(true);

 }

}
```



```
import javax.swing.*;

public class Example{

 public static void main(String[] args){
    JFrame jf=new JFrame("Swing Example");
    jf.setSize(300,300);
    jf.setVisible(true);
    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }

}
```



```
import javax.swing.*;

public class Example extends JFrame{

public Example() { }
public Example(String s)
{
  super(s);
}
 public static void main(String[] args){
   Example jf=new Example("Swing Example");

   jf.setSize(300,300);
   jf.setVisible(true);
   jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }


}
```