

Chapter 7: Java Applets

**Java programs without
main method**

Introduction

- Java language offers an option to write two kinds of programs:
 - Programs that run as a “stand-alone” applications &
 - Programs as applets that are dependent on a controlling program such as a Web browser.
- The big difference between both is:
 - Applications contain enough code to work on their own
 - Applets need a controlling program to tell the applet when to do what.
- An applet has no means of starting execution because it does not have a **main()** method.

Applet Basics

- You need to subclass the **Applet** (Japplet) class.
 - By doing this, you inherit quite a bit of applet functionality that is built-in to the **Applet** class.
- The following listing provides a detailed look at the components of the **Applet** class that are in

```
java.lang.Object
```

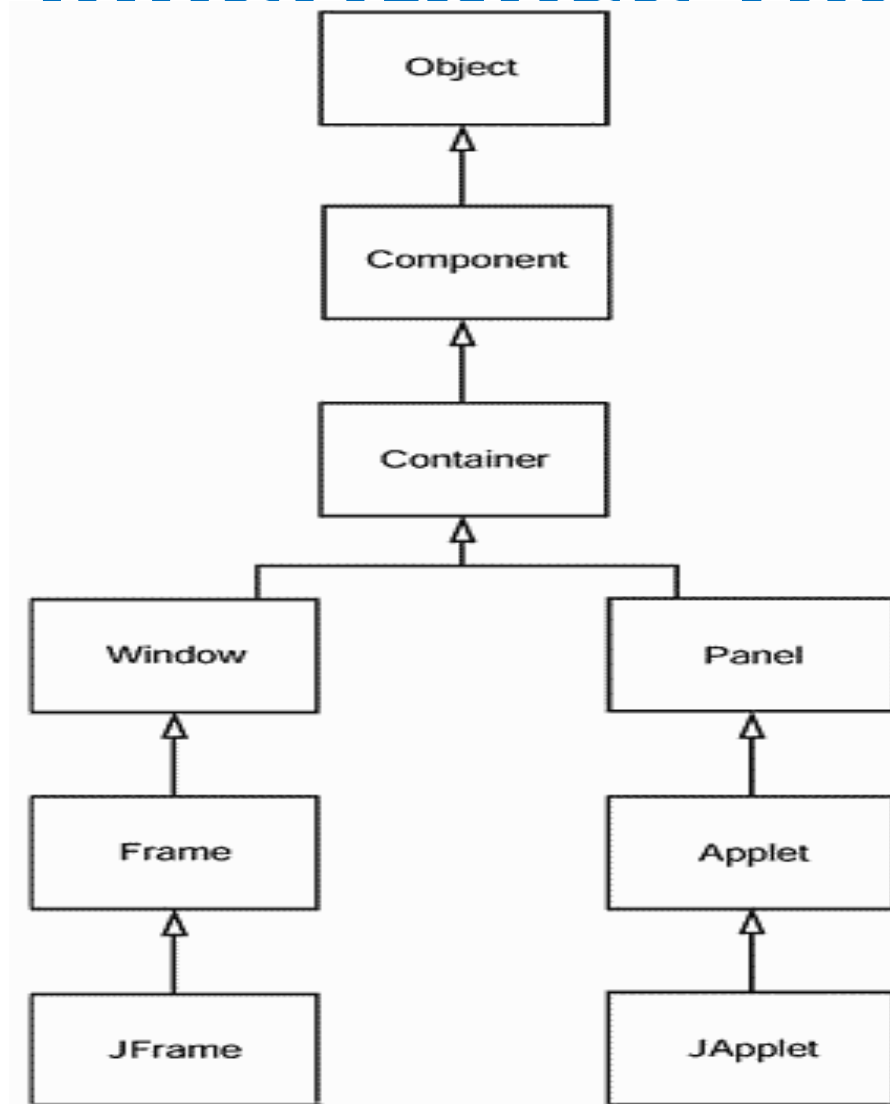
```
    java.awt.Component
```

```
        java.awt.Container
```

```
            java.awt.Panel
```

```
                java.applet.Applet
```

Applet inheritance Hierarchy



Applet basics...

- The **Applet** class is a descendant of the **Container** class;
 - **Thus, it can hold other objects.**
- You do not need to create a panel first to place objects on because the **Applet** class extends the Panel class.
- Because the **Container** class is derived from the **Component** class, we have the ability to respond to events, grab and display images, and display text among many other things.

Applet basics...

- Let us see some of the key methods you have access to because of all the classes that have been extended to get to the **Applet** class.
- **setBackground(Color)**: Sets the background color.
- **setFont(Font)**: Sets the font of the component.
- **setForeground(Color)**: Sets the foreground color.
- **show()**: Shows the component.
- **setColor(Color)**: sets the color for the image to be draw.

Controlling applet life cycle

- Four methods in the Applet class give you the framework on which you build any serious applet: **init**, **start**, **stop**, **destroy**.
- `public void init()`:
 - This method is used for whatever initialization is needed for your applet.
 - This works much like a constructor - it is automatically called by the system when Java launches the applet for the first time.
 - Common actions in an applet include processing PARAM values and adding user interface components.

Controlling applet life...

- `public void start():`
 - This method is automatically called after Java calls the `init` method.
 - It is also called whenever the user returns to the page containing the applet after having gone off to other pages.
 - This means that the `start` method can be called repeatedly, unlike the `init` method.
 - For this reason, put the code that you want executed only once in the `init` method, rather than in the `start` method.
 - The `start` method is where you usually restart a thread for your applet, for example, to resume an animation.

Controlling applet life...

- `public void stop():`
 - This method is automatically called when the user moves off the page on which the applet sits.
 - It can, therefore, be called repeatedly in the same applet.
 - Its purpose is to give you a chance to stop a time-consuming activity from slowing down the system when the user is not paying attention to the applet.
 - You should not call this method directly.
 - If your applet does not perform animation, play audio files, or perform calculations in a thread, you do not usually need this method.

Controlling applet life...

- `public void destroy()`:
 - Java guarantees to call this method when the browser shuts down normally.
 - Since applets are meant to live on an HTML page, you do not need to worry about destroying the panel.
 - This will happen automatically when the browser shuts down.
 - What you do need to put in the destroy method is the code for reclaiming any non-memory-dependent resources such as graphics contexts that you may have consumed.
 - Of course, Java calls the stop method before calling the destroy method if the applet is still active.

Writing your first applet program

- The following program is a simple applet program that displays a welcome message.

```
import java.awt.Graphics;  
import java.applet.*;
```

```
public class WelcomeToApplet extends Applet {  
    public void init() {  
        setBackground(Color.black);  
        setForeground(Color.green);  
    }  
    public void paint(Graphics g){  
        g.drawString("Welcome to applet programming.", 100, 100);  
    }  
}
```

Security basics with applets

- Because applets are designed to be loaded from a remote site and then executed locally, security becomes vital.
- If a user enables Java in the browser, the browser will download all the applet code on the web page and execute it immediately.
- The user never gets a chance to confirm or to stop individual applets from running.
- For this reason, applets (unlike applications) are restricted in what they can do.
- The applet security manager throws a **SecurityException** whenever an applet attempts to violate one of the access rules.

Security basics...

- Applets can do the ff on all platforms:
 - They can show images and play sounds,
 - Get keystrokes and mouse clicks from the user, and
 - Send user input back to the host from which they were loaded.
- The restricted execution environment for applets is often called the "sandbox."
- Applets playing in the "sandbox" cannot alter the user's system or spy on it.

Security basics...

- In particular, when running in the sandbox:
 - Applets can never run any local executable program.
 - Applets cannot communicate with any host other than the server from which they were downloaded; that server is called the originating host.
 - Applets cannot read from or write to the local computer's file system.
 - Applets cannot find out any information about the local computer, except for the Java version used, the name and version of the operating system, and the characters used to separate files (for instance, / or \), paths (such as : or ;), and lines (such as \n or \r\n). In particular, applets cannot find out the user's name, e-mail address, and so on.
 - All windows that an applet pops up carry a warning message.