**INTRODUCTION ON APPLETS**

**Applet Basics**

* Applets are Java programs that are integrated into Web pages. When a Web page containing an applet is displayed by a Web browser, the applet is loaded and executed. The applet's output is displayed within a subset of the browser's display area.
* All applets are subclasses of Applet. Thus, all applets must import java.applet. Applets must also import java.awt. AWT stands for the Abstract Window Toolkit. Since all applets run in a window, it is necessary to include support for that window.
* Applets are executed by either a Web browser or an applet viewer
* Execution of an applet does not begin at main( ). Execution of an applet is started and controlled with an entirely different mechanism. Output to your applet’s window is not performed by System.out.println( ). Rather, it is handled with various AWT methods, such as drawString( ), which outputs a string to a specified X,Y Location.
* Once an applet has been compiled, it is included in an HTML file using the APPLET tag. The applet will be executed by a Java-enabled web browser when it encounters the APPLET tag within the HTML file. To view and test an applet more conveniently, simply include a comment at the head of your Java source code file that contains the APPLET tag..
* Here is an example of such a comment:

/\*

<applet code="MyApplet" width=200 height=60>

</applet>

\*/

### The Applet Class

* The java.applet package is one of the smallest packages in the Java API. It consists of a single class, the Applet class, and three interfaces: AppletContext, AppletStub, and AudioClip.
* The Applet class contains a single default parameterless constructor, which is generally not used. Applets are constructed by the runtime environment when they are loaded and do not have to be explicitly constructed.

### How Applets and Applications Are Different

* In short, Java applications are stand-alone Java programs that can be run by using just the Java interpreter, for example, from a command line Java applets, however, are run from inside a World Wide Web browser. A reference to an applet is embedded in a Web page using a special HTML tag
* Because Java applets run inside a Java browser, they have the advantage of the structure the browser provides: an existing window, an event-handling and graphics context, and the surrounding user interface. Java applications can also create this structure, but they don't require it
* The convenience that applets have over applications in terms of structure and UI capabilities, however, is hampered by restrictions on what applets can do. Given the fact that Java applets can be downloaded from anywhere and run on a client's system, restrictions are necessary to prevent an applet from causing system damage or security breaches. Without these restrictions in place, Java applets could be written to contain viruses or trojan horses (programs that seem friendly but do some sort of damage to the system), or be used to compromise the security of the system that runs them.

**What applets can’t do?**

* Applets can't read or write to the reader's file system, except in specific directories (which are defined by the user through an access control list that, by default, is empty). Some browsers may not even allow an applet to read or write to the file system at all, or at the same time as using the network.
* Applets can't usually communicate with a server other than the one that had originally stored the applet. (This may be configurable by the browser; however, you should not depend on having this behavior available.)
* Applets can't run any programs on the reader's system. For Unix systems, this includes forking a process.
* Applets can't load programs native to the local platform, including shared libraries such as DLLs.

### Creating Applets

* To create an applet, you create a subclass of the class Applet, in the java.applet package. The Applet class provides behavior to enable your applet not only to work within the browser itself, but also to take advantage of the capabilities of AWT to include UI elements, to handle mouse and keyword events, and to draw to the screen.
* Although your applet can have as many "helper" classes as it needs, it's the main applet class that triggers the execution of the applet. That initial applet class always has a signature like this:

public class myClass extends java.applet.Applet {

...

}

* Note the public keyword. Java requires that your applet subclass be declared public. Again, this is true only of your main applet class; any helper classes you create can be public or private as you wish.
* When Java encounters your applet in a Web page, it loads your initial applet class over the network, as well as any other helper classes that first class uses. Unlike with applications, where Java calls the main() method directly on your initial class, when your applet is loaded, Java creates an instance of that class, and a series of Applet methods are called on that instance.
* Different applets that use the same class use different instances, so each one can behave differently from the other applets running in the same browser.

#### Applet Life Cycle

* **init()**

Called when the applet is first loaded into the browser or viewer. It is typically used to perform applet initialization, in preference to a constructor method. (The Web browser doesn't pass any arguments to an applet's constructor method, so defining one isn't too useful.)

* **destroy()**

Called when the applet is about to be unloaded from the browser or viewer. It should free any resources, other than memory, that the applet has allocated.

* **start()**

Called when the applet becomes visible and should start doing whatever it is that it does. Often used with animation and with threads.

* **paint()**

Painting is how an applet actually draws something on the screen, be it text, a line, a colored background, or an image. You override the paint() method for your applet to have an actual appearance on the screen. The paint() method looks like this:

public void paint(Graphics g) {

...

}

* **stop()**

Called when the applet becomes temporarily invisible, for example, when the user has scrolled it off the screen. Tells the applet to stop performing an animation or other task.

* Note that unlike the other major methods in this section, paint() takes an argument, an instance of the class Graphics. This object is created and passed to paint by the browser, so you don't have to worry about it. However, you will have to make sure that the Graphics class (part of the java.awt package) gets imported into your applet code, usually through an import statement at the top of your Java file: