UNIT - V

Applet Programming *and* *The Java IO System*

Applet Programming and The Java IO System

Applet Programming: Introduction, How Applets Differ from Applications, Preparing to Write Applets, Building Applet Code, Applet Life Cycle, Creating an Executable Applet, Designing a Web Page, Applet Tag, Adding Applet to HTML File, Running the Applet, More About Applet Tag, Passing Parameters to Applets, Aligning the Display, More About HTML Tags, Displaying Numerical Values, Getting Input from the User, Event Handling.

Managing Input/Output Files in Java: *Introduction, Concept of Streams, Stream Classes, Byte Stream Classes, Character Stream Classes, Using Streams, Other Useful I/O Classes, Using the File Class, Input/Output Exceptions, Creation of Files, Reading/Writing Characters, Reading/Writing Bytes, Handling Primitive Data Types, Concatenating and Buffering Files, Random Access Files, Interactive Input and Output, Other Stream Classes.*

Applet Programming

- Introduction, How Applets Differ from Applications, Preparing to Write Applets, Building Applet Code, Applet Life Cycle, Creating an Executable Applet, Designing a Web Page, Applet Tag, Adding Applet to HTML File, Running the Applet, More About Applet Tag, Passing Parameters to Applets, Aligning the Display, More About HTML Tags, Displaying Numerical Values, Getting Input from the User, Event Handling.

Applet Programming

Introduction

Applet Fundamentals

- Applets are small applications that are accessed on an Internet server, transported over the Internet, automatically installed, and run as part of a web document.
- After an applet arrives on the client, it has limited access to resources so that it can produce a graphical user interface and run complex computations without introducing the risk of viruses or breaching data integrity.

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet:***

```
import java.awt.*;
import java.applet.*;
public class SimpleApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("A Simple Applet", 20, 20);
    }
}
```

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- This applet begins with two **import** statements. The first imports the Abstract Window Toolkit (AWT) classes. Applets interact with the user (either directly or indirectly) through the AWT, not through the console-based I/O classes. The AWT contains support for a window-based, GUI. Fortunately, this simple applet makes very limited use of the AWT. (Applets can also use Swing to provide the GUI.)
- The second import statement **imports** the **applet** package, which contains the class **Applet**. Every applet that you create must be a subclass of **Applet**.
- The next line in the program declares the class **SimpleApplet.** This class must be declared as **public**, because it will be accessed by code that is

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- Inside **SimpleApplet**, **paint( )** is declared. This method is defined by the AWT and must be overridden by the **applet**. **paint( )** is called each time that the applet must redisplay its output. whenever the applet must redraw its output, **paint( )** is called.
- The **paint( )** method has one parameter of type **Graphics**. This parameter contains the graphics context, which describes the graphics environment in which the applet is running. This context is used whenever output to the applet is required.
- Inside **paint( )** is a call to **drawString( )**, which is a member of the Graphics class. This method outputs a string beginning at the specified X,Y location.

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- The general form: **void drawString(String message, int x, int y)**
- Here, message is the string to be output beginning at x,y. In a Java window, the upper-left corner is location 0,0. The call to **drawString( )** in the applet causes the message "A Simple Applet" to be displayed beginning at location 20,20.
- Notice that the applet does not have a **main( )** method. Unlike Java programs, applets do not begin execution at **main( )**. In fact, most applets don't even have a **main( )** method. Instead, an applet begins execution when the name of its class is passed to an applet viewer or to a network browser.

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- After you enter the source code for **SimpleApplet**, compile in the same way that you have been compiling programs. However, running SimpleApplet involves a different process. In fact, there are two ways in which you can run an applet:
  - Exceuting the applet within a Java-compatible web browser.
  - Using an applet viewer, such as the standard tool, ***appletviewer***. An applet viewer executes your applet in a window. This is generally the fastest and easiest way to test your applet.

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- To execute an applet in a web browser, you need to write a short HTML text file that contains a tag that loads the applet. Currently, Sun recommends using the APPLET tag for this purpose. Here is the HTML file that executes **SimpleApplet**:

```
<applet code="SimpleApplet" width=200 height=60>
</applet>
```

- The **width** and **height** statements specify the dimensions of the display area used by the applet. (The APPLET tag contains several other options.) After you create this file, you can execute your browser and then load this file, which causes **SimpleApplet** to be executed.

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- To execute **SimpleApplet** with an applet viewer, you may also execute the HTML file shown earlier. For example, if the preceding HTML file is called **RunApp.html**, then the following command line will run **SimpleApplet**:

```
C:\>appletviewer RunApp.html
```

Applet Programming

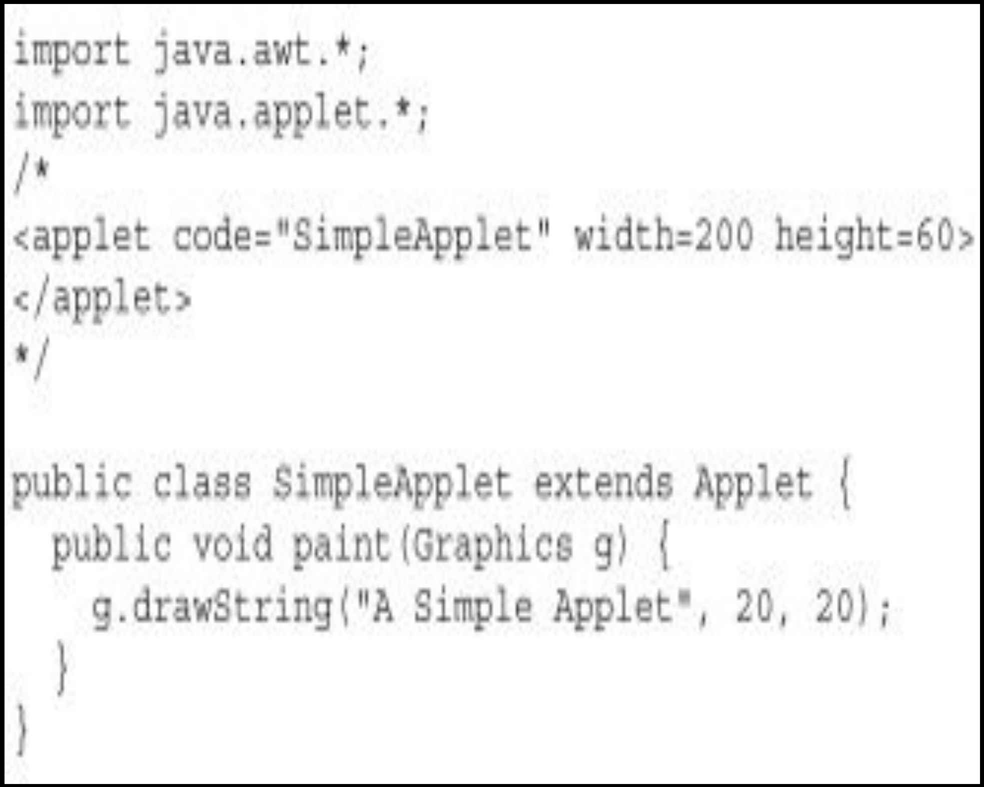
Introduction ...

Applet Fundamentals

- ***The simple applet: ...***

However, a more convenient method exists that you can use to speed up testing. Simply include a comment at the head of your Java source code file that contains the APPLET tag.

By doing so, your code is documented with a prototype of the necessary HTML statements, and you can test your compiled applet merely by starting the applet viewer with your Java source code file. If you use this method, the **SimpleApplet** source file looks like this:



```
import java.awt.*;
import java.applet.*;
/*
<applet code="SimpleApplet" width=200 height=60>
</applet>
*/

public class SimpleApplet extends Applet {
  public void paint(Graphics g) {
    g.drawString("A Simple Applet", 20, 20);
  }
}
```

Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***
- With this approach, we can quickly iterate through applet development by using these three steps:

1) Edit a Java source file.
2) Compile your program.
3) Execute the applet viewer, specifying the name of your applet's source file. The applet viewer will encounter the APPLET tag within the comment and execute your applet.

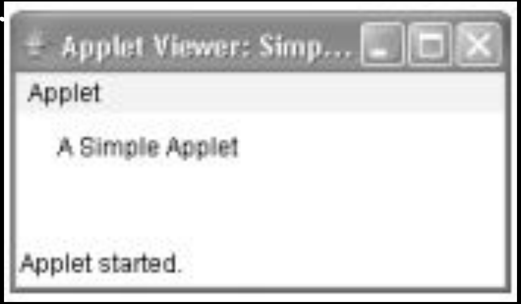
Applet Programming

Introduction ...

Applet Fundamentals

- ***The simple applet: ...***

- The window produced by **SimpleApplet,** as displayed by the applet viewer, is shown in the following illustr



A Simple Applet

Applet started.

- **The key points that you should remember now:**
  - Applets do not need a **main( )** method.
  - Applets must be run under an applet viewer or a Java-compatible browser.
  - User I/O is not accomplished with Java's stream I/O classes. Instead, applets use the interface provided by the AWT or Swing.

Applet Programming

Introduction

- Java programs are divided into two main categories, applets and applications.
- An application is an ordinary Java program.
- An applet is a kind of Java program that can be run across the Internet.
- Applets are small Java programs that are embedded in Web pages.
- They can be transported over the Internet from one computer (web server) to another (client computers).
- They transform web into rich media and support the delivery of applications via the Internet.

Applet Programming

Introduction ...

- All applets are subclasses (either directly or indirectly) of **Applet**. Applets are not stand-alone programs. Instead, they run within either a web browser or an applet viewer.
- The illustrations shown in this chapter were created with the standard applet viewer, called **appletviewer**, provided by the **JDK**.
- But you can use any applet viewer or browser you like. Execution of an applet does not begin at **main( ).** Actually, few applets even have **main( )** methods.
- Instead, execution of an applet is started and controlled with an entirely different mechanism.

Applet Programming

Advantages

- There are many advantages:
  - It works at client side so less response time.
  - Secured
  - It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.

Drawback

- Plug-in is required at client browser to execute applet.

Applet Programming

How Applets Differ from Applications

- Although both the Applets and stand-alone applications are Java programs, there are certain restrictions are imposed on Applets due to security concerns:
    - Applets don’t use the main() method, but when they are load, automatically call certain methods (init, start, paint, stop, destroy).
    - They are embedded inside a web page and executed in browsers.
    - They cannot read from or write to the files on local computer.
    - They cannot communicate with other servers on the network.
    - They cannot run any programs from the local computer.
    - They are restricted from using libraries from other languages.
- The above restrictions ensures that an Applet cannot do any damage to the local system.

Applet Programming

Building Applet Code: *An Example*

```
//HelloWorldApplet.java
import java.applet.Applet;
import java.awt.*;

public class HelloWorldApplet extends Applet {
        public void paint(Graphics g) {
                g.drawString ("Hello World of Java!",25, 25);
        }
}
```

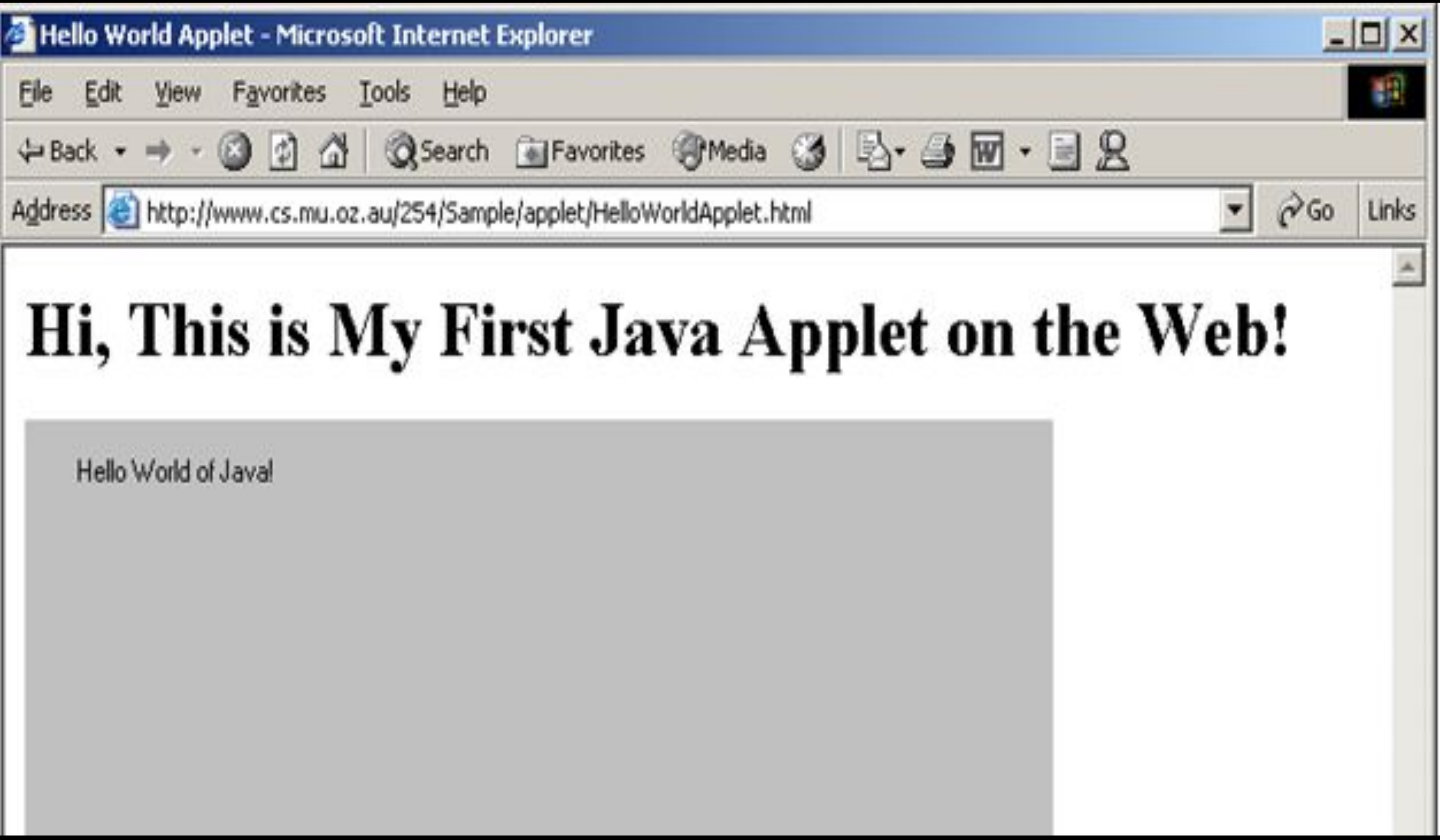
Applet Programming

Embedding Applet in Web Page

```
<HTML>
            <HEAD>
                  <TITLE>
                          Hello World Applet
                  </TITLE>
          </HEAD>
      <body>
            <h1>Hi, This is My First Java Applet on the Web!</h1>
            <APPLET CODE="HelloWorldApplet" width=500 height=400>
            </APPLET>
      </body>
</HTML>
```

Applet Programming

Accessing Web page (runs Applet)


Hello World Applet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.cs.mu.oz.au/254/Sample/applet/HelloWorldApplet.html
Go Links

Hi, This is My First Java Applet on the Web!

```
Hello World of Java!
```

Applet Programming

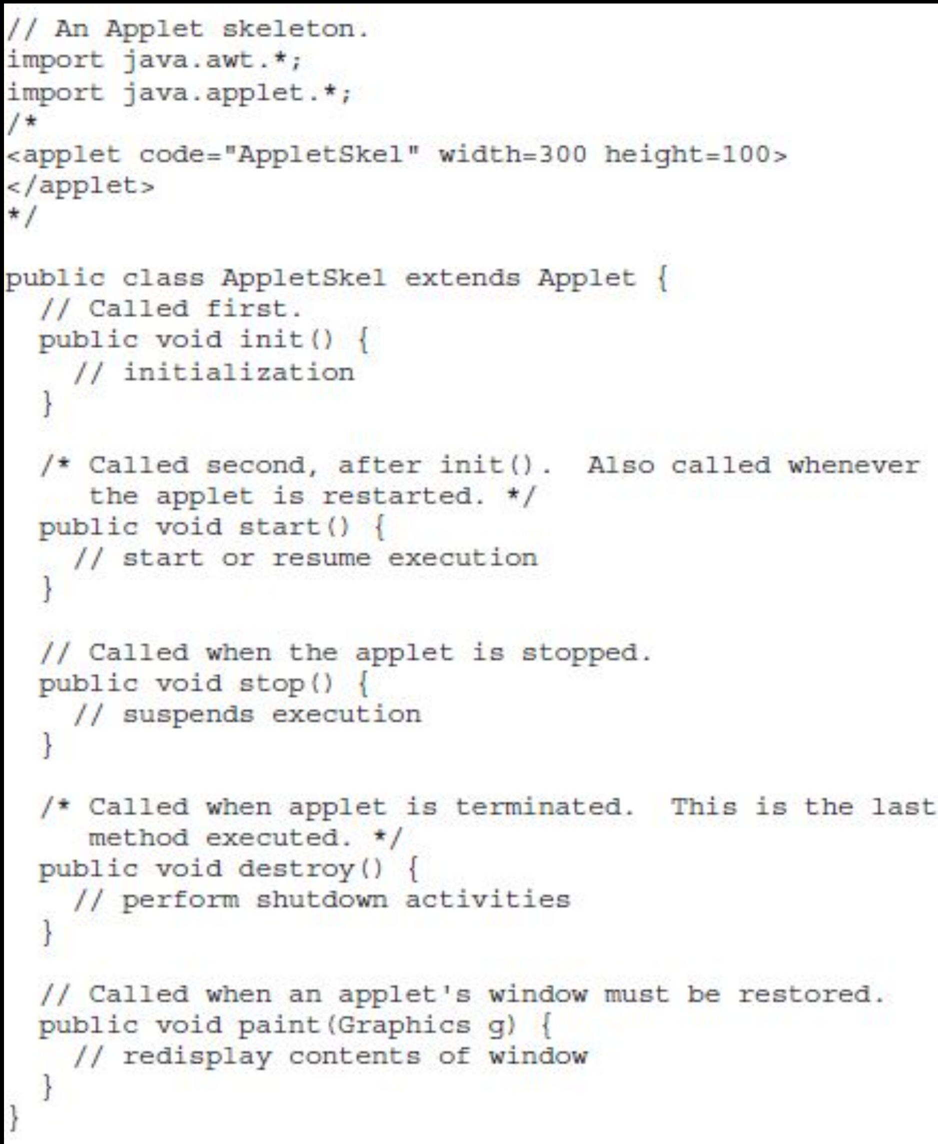
An Applet Skeleton / Applet Life Cycle

- All but the most trivial applets override a set of methods that provides the basic mechanism by which the browser or applet viewer interfaces to the applet and controls its execution.
- Four of these methods, **init( ), start( ), stop( ), and destroy( )**, apply to all applets and are defined by **Applet**.
- Default implementations for all of these methods are provided. Applets do not need to override those methods they do not use. However, only very simple applets will not need to define all of them.

Applet Programming

An Applet Skeleton / Applet Life Cycle ...

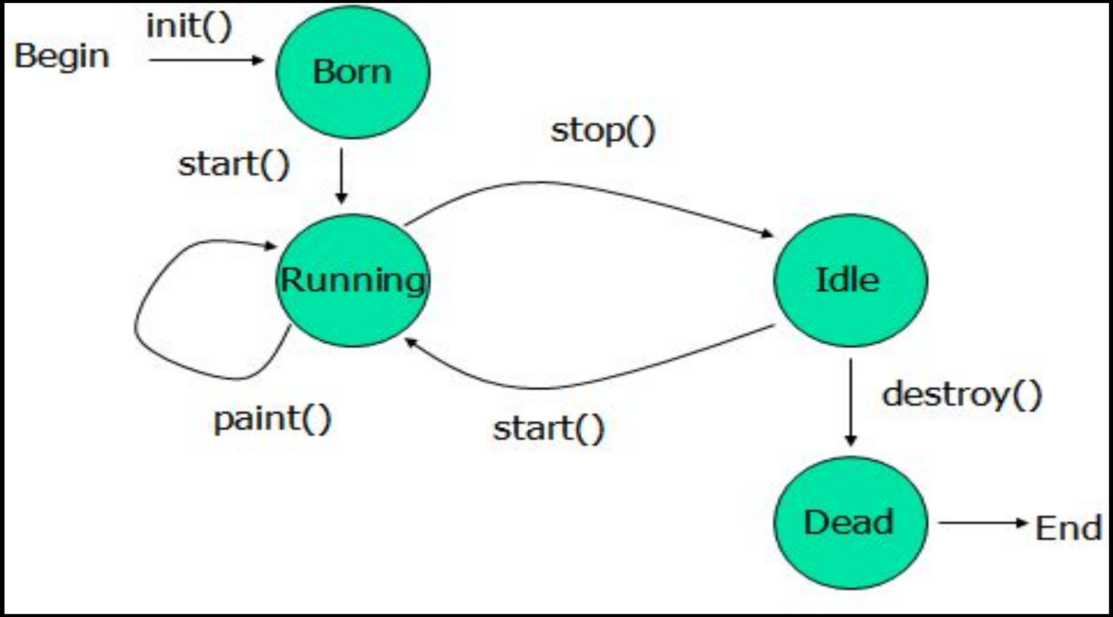
- Every applet inherits a set of default behaviours from the **Applet** class. As a result, when an applet is loaded, it undergoes a series of changes in its state. The applet states include:
  - **Initialisation – invokes init()**
  - **Running – invokes start()**
  - **Display – invokes paint()**
  - **Idle – invokes stop()**
  - **Dead/Destroyed State – invokes destroy()**



```
// An Applet skeleton.
import java.awt.*;
import java.applet.*;
/*
<applet code="AppletSkel" width=300 height=100>
</applet>
*/

public class AppletSkel extends Applet {
  // Called first.
  public void init() {
    // initialization
  }

  /* Called second, after init().  Also called whenever
     the applet is restarted. */
  public void start() {
    // start or resume execution
  }

  // Called when the applet is stopped.
  public void stop() {
    // suspends execution
  }

  /* Called when applet is terminated.  This is the last
     method executed. */
  public void destroy() {
    // perform shutdown activities
  }

  // Called when an applet's window must be restored.
  public void paint(Graphics g) {
    // redisplay contents of window
  }
}
```

Applet Programming

An Applet Skeleton / Applet Life Cycle ...



- **public void init():** is used to initialized the Applet. It is invoked only once.
- **public void start():** is invoked after the init() method or browser is maximized. It is used to start the Applet.
- **public void stop():** is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
- **public void destroy():** is used to destroy the Applet. It is invoked only once.

Applet Programming

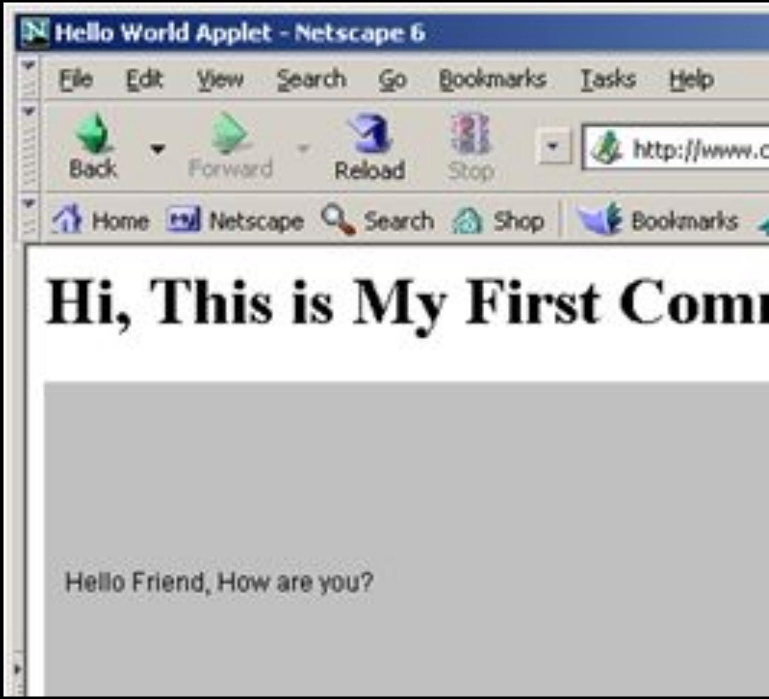
Passing Parameters to Applet

```
<HTML>

    <HEAD>
        <TITLE>
            Hello World Applet
        </TITLE>
    </HEAD>

    <body>
    <h1>Hi, This is My First Communicating Applet on the Web!</h1>
        <APPLET CODE="HelloAppletMsg.class" width=500 height=400>
          <PARAM NAME="Greetings" VALUE="Hello Friend, How are you?">
        </APPLET>

    </body>

</HTML>
```

Applet Programming

Applet Program Accepting Parameters

```
//HelloAppletMsg.java
import java.applet.Applet;
import java.awt.*;

public class HelloAppletMsg extends Applet {

    String msg;

    public void init()
    {
        msg = getParameter("Greetings");
        if( msg == null)
            msg = "Hello";
    }
    public void paint(Graphics g) {
        g.drawString (msg,10, 100);
    }
}
```


Hello World Applet - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
Back Forward Reload Stop
http://www.c
Home Netscape Search Shop Bookmarks

Hi, This is My First Com

Hello Friend, How are you?

This is name of parameter specified in PARAM tag; This method returns the value of paramter.

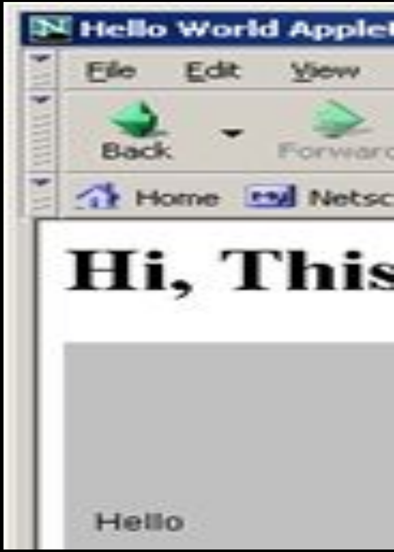
Applet Programming

What happen if we don't pass parameter?

See HelloAppletMsg1.html

```
<HTML>

    <HEAD>
        <TITLE>
            Hello World Applet
        </TITLE>
    </HEAD>

    <body>
        <h1>Hi, This is My First Communicating Applet on the Web!</h1>
        <APPLET CODE="HelloAppletMsg.class" width=500 height=400>
        </APPLET>

    </body>

</HTML>
```

getParameter() returns *null*. Some default value may be used.



Hi, This

Hello

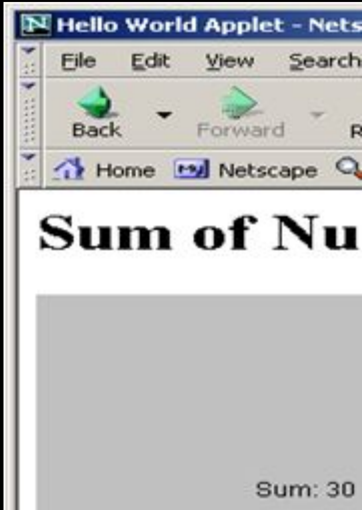
Applet Programming

Displaying Numeric Values

```
//SumNums.java
import java.applet.Applet;
import java.awt.*;

public class SumNums extends Applet {
   public void paint(Graphics g) {
      int num1 = 10;
      int num2 = 20;
      int sum = num1 + num2;
      String str = "Sum: "+String.valueOf(sum);
      g.drawString (str,100, 125);
   }
}
```

SumNums.html

```
<HTML>

<HEAD>
        <TITLE>
                Hello World Applet
        </TITLE>
</HEAD>

<body>
<h1>Sum of Numbers</h1>
<APPLET CODE="SumNums.class" width=500
height=400>
</APPLET>
</body>

</HTML>
```



Sum of Nu

Sum: 30

Applet Programming

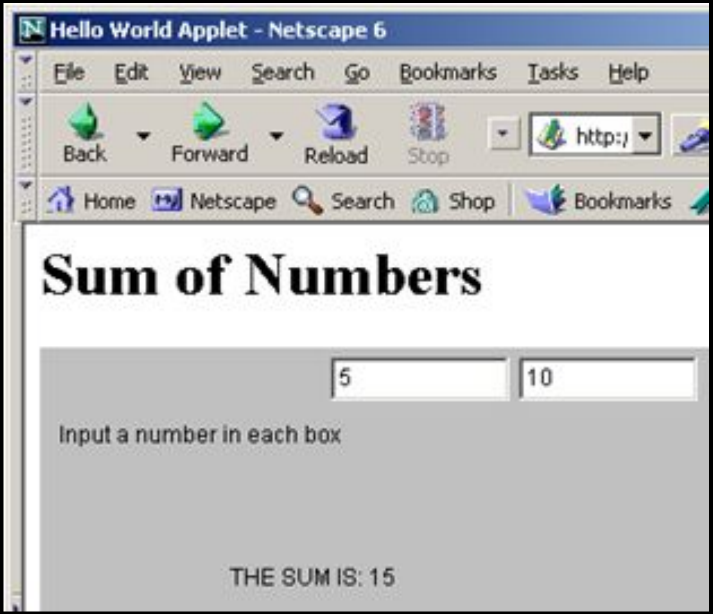
Interactive Applet

- Applets work in a graphical environment. Therefore, applets treats inputs as text strings.
- We need to create an area on the screen in which use can type and edit input items.
- We can do this using TextField class of the applet package.
- When data is entered, an event is generated. This can be used to refresh the applet output based on input values.

Applet Programming

Interactive Applet ...

```
//SumNumsInteractive..java
import java.applet.Applet;
import java.awt.*;

public class SumNumsInteractive extends Applet {
        TextField text1, text2;
        public void init()
        {
          text1 = new TextField(10);
          text2 = new TextField(10);
          text1.setText("0");
          text2.setText("0");
          add(text1);
          add(text2);
        }
        public void paint(Graphics g) {
          int num1 = 0;
          int num2 = 0;
          int sum;
          String s1, s2, s3;

          g.drawString("Input a number in each box ", 10, 50);
          try {
                s1 = text1.getText();
                num1 = Integer.parseInt(s1);
                s2 = text2.getText();
                num2 = Integer.parseInt(s2);
          }
          catch(Exception e1)
          {}
          sum = num1 + num2;
          String str = "THE SUM IS: "+String.valueOf(sum);
          g.drawString (str,100, 125);
        }
        public boolean action(Event ev, Object obj)
        {
                repaint();
                return true;
        }
}
```


Hello World Applet - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
Back Forward Reload Stop
Home Netscape Search Shop Bookmarks

Sum of Numbers

5 10

Input a number in each box

THE SUM IS: 15

Applet Programming

Applet and Security

- An applet can be a program, written by someone else, that runs on your computer.
- Whenever someone else's program runs on your computer, there are security questions you should ask:
  - Will it read information from your files?
  - Will it corrupt your operating system?
- Applets are designed so that they cannot do any of these things (at least easily).

Applet Programming

Summary

- Applets are designed to operate in Internet and Web environment.
- They enable the delivery of applications via the Web.
- In this presentation we learned:
  - ✔ *How do applets differ from applications?*
  - ✔ *Life cycles of applets*
  - ✔ *How to design applets?*
  - ✔ *How to execute applets?*
  - ✔ *How to provide interactive inputs?*

Applet Programming

Additional Resources:

Applet Programming-1

Applet Programming-2

Applet Programming-3