

Create a Java Applet

Those of you who purchased my latest book, Learn to Program with Java, know that in the book, we create a Java program designed to calculate grades for the English, Math and Science departments of a university. At the completion of the book, we have a working Java program that uses a windows Interface to prompt for grade components, and then displays the results. In this article, I'm going to show you how to take the project and modify it so that it can be displayed in a web page. I think you'll find the process amazingly painless.

What's an Applet?

An applet is just a Java class that runs in a Web Browser such as Internet Explorer or Netscape. Creating an Applet is really easy, unfortunately, there's one problem. Neither Internet Explorer or Netscape support the latest and greatest from Java--namely its Swing Package, so if you intend to use Swing components in your Applet we need to do a little more work. But let's start with a simple Applet so that you can see what's going on.

Let's write a simple applet from Scratch

Here's the code for our first Applet which will display 'I love Java' as a message in a Web page

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

public class Example14_1 extends Applet {
    public void paint(Graphics g) {
        g.drawString("I love Java",20,20);
    }
}
```

Let's take a closer look at the code and then we'll compile it and discuss how to 'run' it in a Web page.

The first thing to note are the two import statements. We need to import 'awt' in order to produce the window within our browser. The second import statement is necessary because all Applets extend, or inherit from, the Java applet class.

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

Because of that, the line of code defining the Applet class must use the keyword 'extends' as this one does here...

```
public class Example14_1 extends Applet {
```

The second thing to note is that Java Applets, unlike the Java Applications we followed in the book, may not have a main() method. A Java Applet, when it fires up in a Web Browser, automatically looks to execute three methods in this order.

- init()
- start()
- paint

You can choose which of these to code. The init() method is executed when an Applet object is first instantiated. The start() method is executed after the init() method completes execution and every time the user of the browser returns to the HTML page on which the applet resides. The paint() method is called after the init method has completed and after the start() method has begun--most importantly, code in the paint() method is guaranteed to run each time the window needs to be re-drawn (i.e. after the browser has been minimized, covered, the HTML page has changed, etc).

For our purposes, we coded the paint method, which uses a Graphics object (here defined as 'g') to do the drawing...

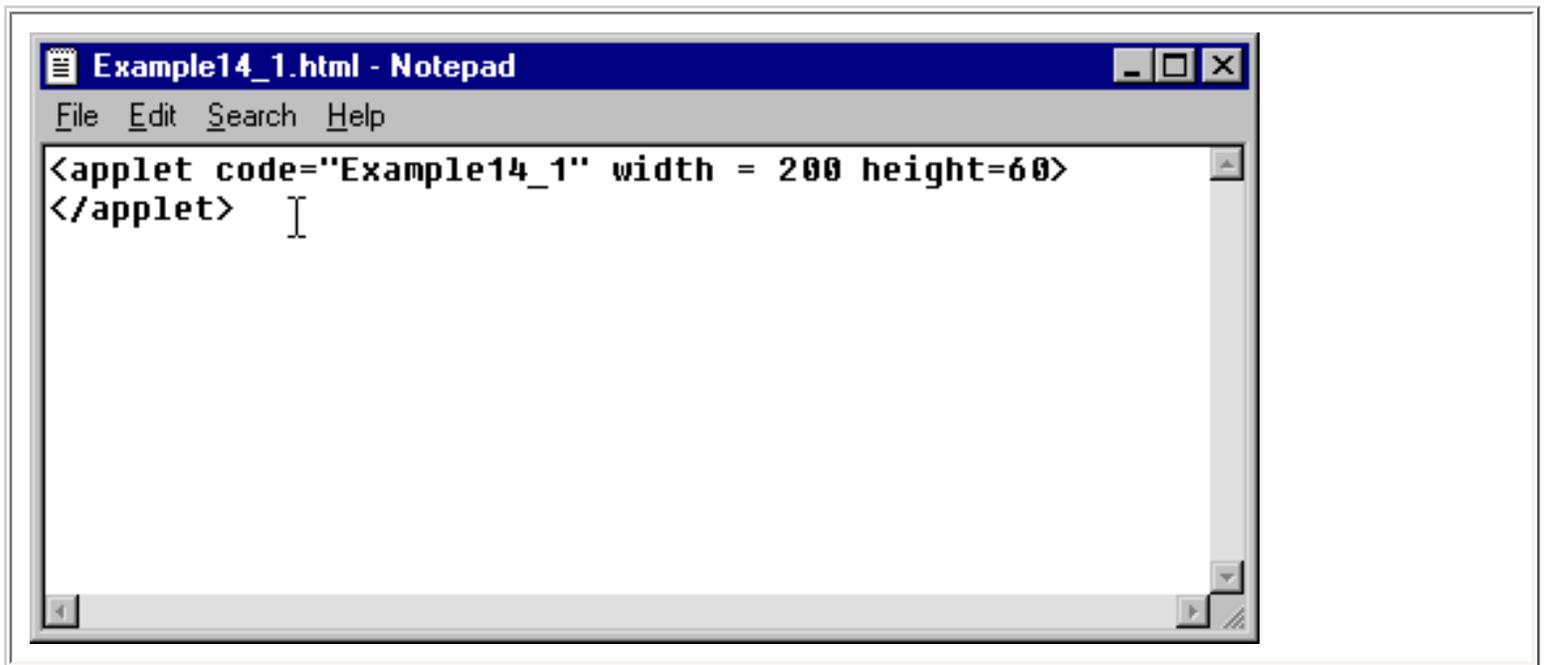
```
public void paint(Graphics g) {
```

The code in the paint() method is used to draw the window that will be displayed in the Web Browser. To do that we use the drawString() method of the Graphics object, supplying the text that we wish to appear in the window and it's dimensions...

```
g.drawString("I love Java",20,20);
```

We need to write some HTML

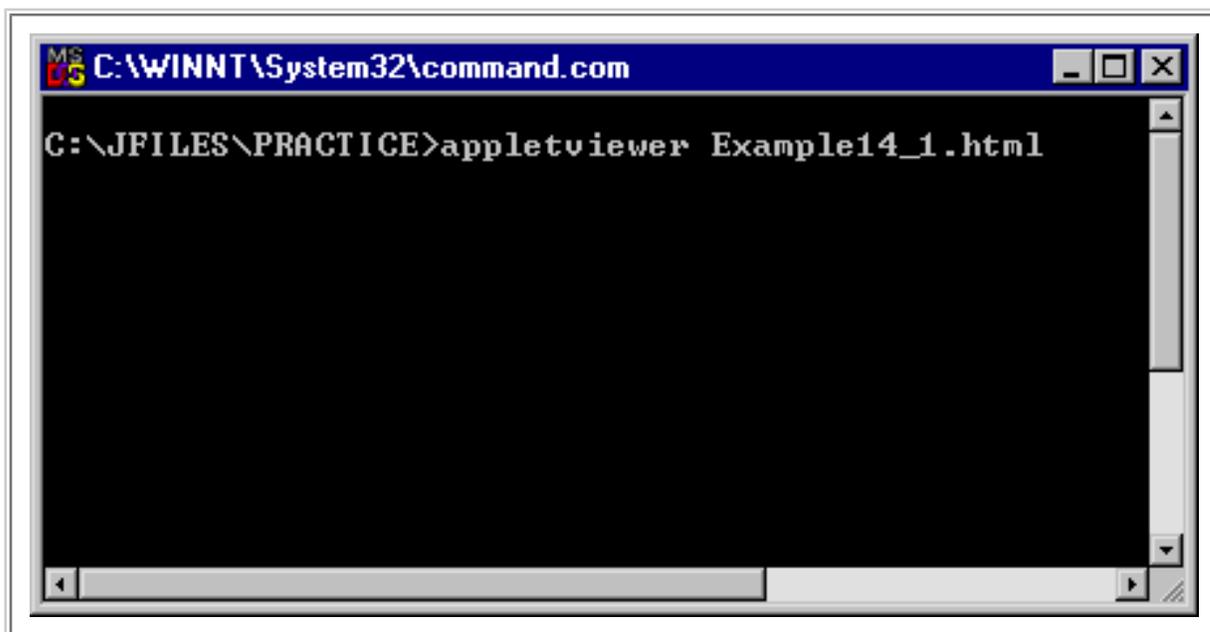
As I mentioned, Applet classes need to run within a Web Browser. Assuming you have now compiled the Java source file into a Bytecode file, we now need to create an HTML document which will create an instance of our Applet class when displayed in a Web Browser. (I don't intend to teach you HTML in this article--there are plenty of books and resources to do that.). Here's the HTML necessary to create an instance of our object and display its output in a Web Browser. I've saved the HTML as Example14_1.html



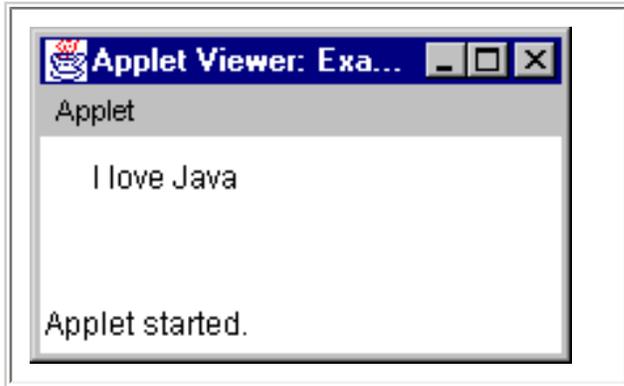
The HTML file contains two lines of code that will, in a bare-bones way, do the trick. HTML works primarily with tags, and the code that you see is an Applet tag to tell the Browser reading the HTML to create an instance of the Example14_1 class, within a window 200 pixels wide and 60 pixel high. We're now ready to load our Applet into a browser, but before we do that, I want to tell you about the Applet Viewer.

Use the Applet Viewer to test our Applet

Java includes, in its Java Developers Kit, a special interpreter called the Applet Viewer which can be used to test the Applets we write. To execute it, open up a Command Window and enter the following instructions to execute the HTML file we just wrote.



If we now press the ENTER key, we should see the Applet Viewer display window, and within it, the results of the execution of our Java applet...

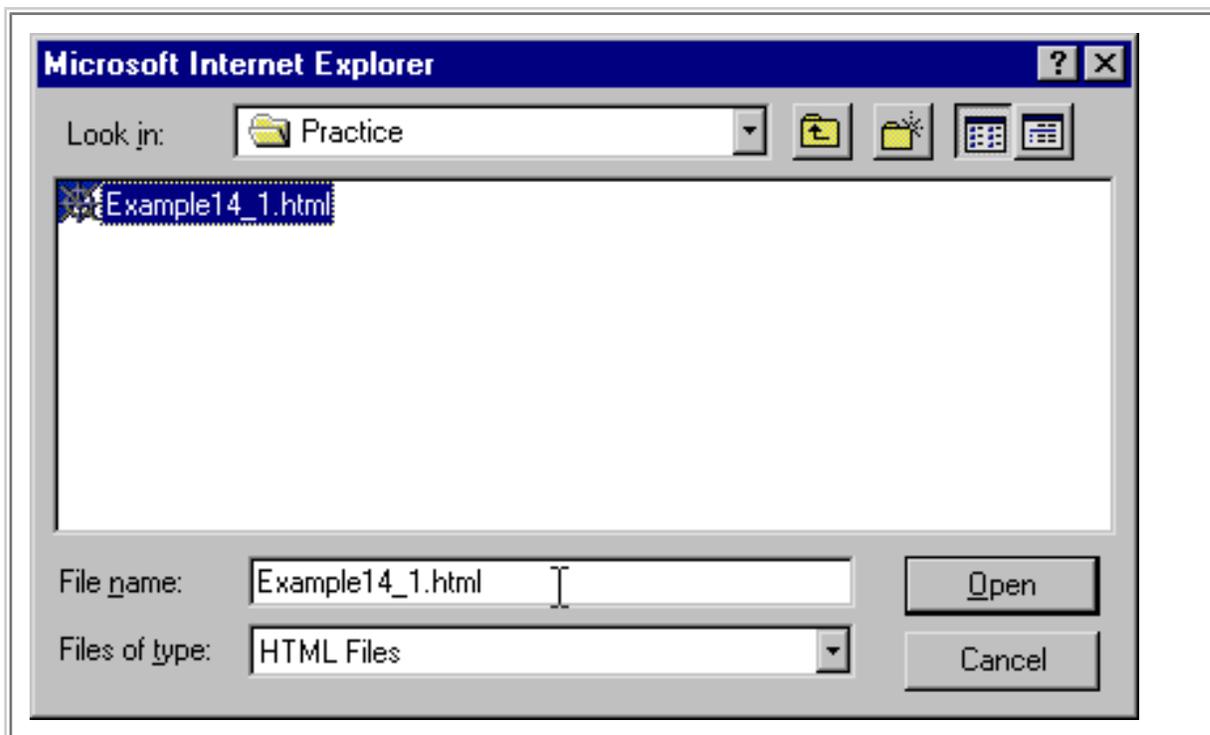


The Applet Viewer can be shut down simply by clicking on its close button or Control Menu Icon.

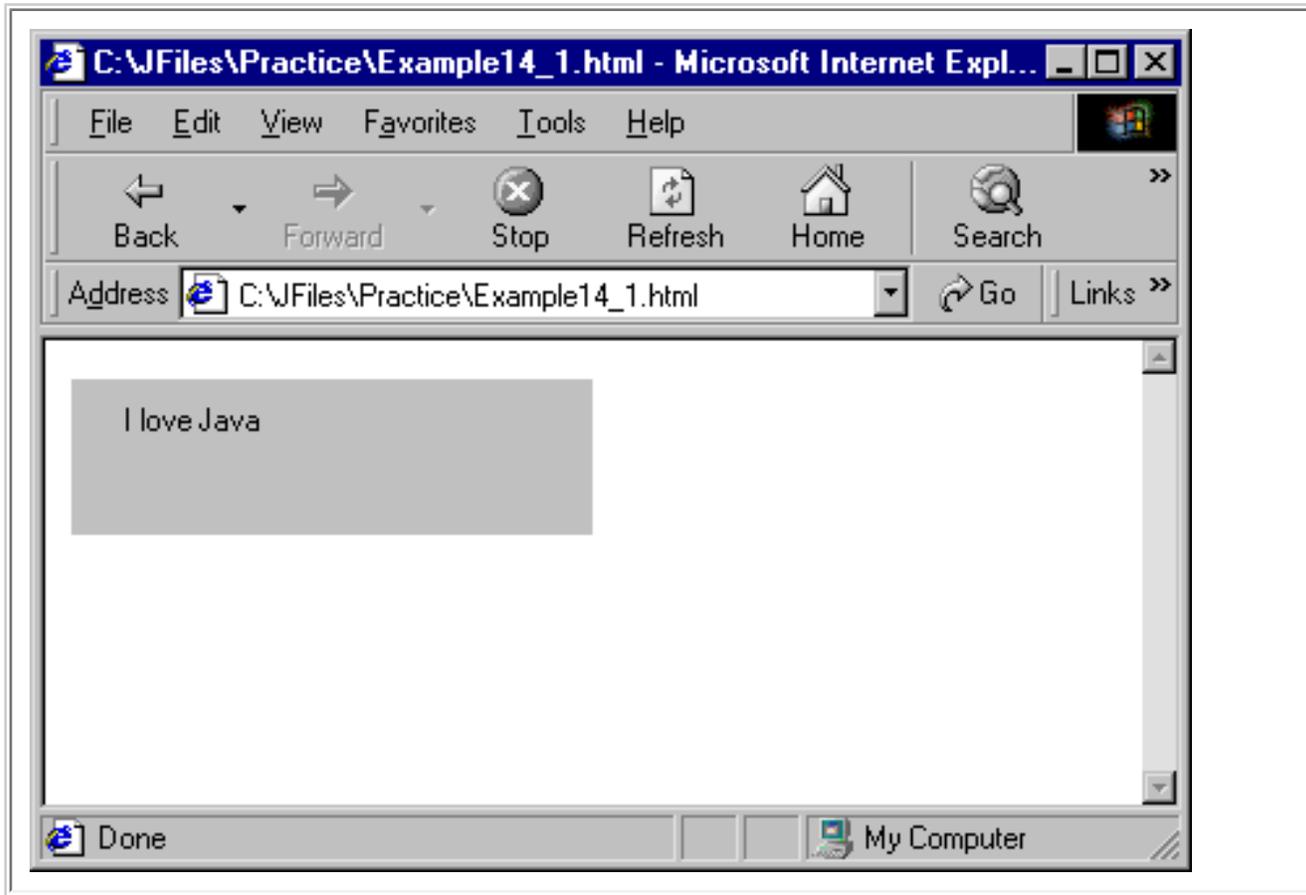
Display our Applet in a Web Browser

To test our Applet in a Web Browser, we need to be sure that our Web Browser supports Java. Both Internet Explorer and Netscape have settings to turn Java 'on'---if you have doubts as to how to do that, check your Browser's help file.

Assuming that your Browser has Java turned 'on', load up the HTML file we created into the Browser. Both Internet Explorer and Netscape permit you to display local HTML pages by selecting File-Open from the Menu Bar and selecting the HTML page. I'll do that here with Internet Explorer (just to show you that Microsoft has nothing against running a Java program!)



Here's our Applet running in Internet Explorer...



As you can see, the results are basically the same as those we found when running within the Applet Viewer---the difference is that the Web Browser is now acting as the container for our object, not the Applet Viewer.

Now it's time to turn our attention to the Grades Calculation Project from my Java book.

Convert the Grades Calculation Project into an Applet

Let's take a look at the code we have right now in the Grades Calculation project. As you may recall, the final Grades project from the end of the book consists of a number of class modules.

- CalculateButtonListener
- DisplayGrade
- DrawGUI
- EnglishStudent
- GradesGUI
- MathStudent
- RadioButtonListener
- ResetButtonListener
- ScienceStudent
- Student

The good news is that in order to run our Java project from within a Web Page, we need to change just one of these classes---the GradesGUI class. Here's the code that currently resides there...

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class GradesGUI {
    public static void main(String args[])
    {
        DrawGUI x = new DrawGUI();
        x.addWindowListener(
            new WindowAdapter() {
                public void windowClosing ( WindowEvent e ) {
                    System.exit ( 0 );
                }
            } // end of windowadpater
        ); // end of windowlistener
    } // end of main
} // END OF CLASS
```

and here's the modified code...

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class GradesApplet extends JApplet {
    public void init()
    {
        DrawGUI x = new DrawGUI();
        x.addWindowListener(
            new WindowAdapter() {
                public void windowClosing ( WindowEvent e )
                {
                    setVisible(false);
                }
            } // end of windowClosing
        ); // end of windowAdpater
    } // end of windowListener
} // end of void
} // end of class
```

The good news here is that most of the code from GradesGUI hasn't changed a bit--and neither have the other nine classes. Java truly lives up to its reputation of permitting us to build hardware and operating system independent class modules. In an ideal world we wouldn't have to change a thing,

but we do need to make a few changes.

I've made three changes here. First, the Class definition now extends the JApplet class. JApplet is used in classes that contain Swing components---think of it as a Swing enabled version of the Applet class we used in the previous example. So that you don't confuse this code with the code you already have from the end of my Java book, let's name this class GradesApplet...

```
public class GradesApplet extends JApplet {
```

The second change is to eliminate the main() method we had previously, and replace it with the init() method, which is its Applet counterpart. Remember, Applets may not have a main() method. We're using Init() here because this class, as did GradesGUI, really just instantiates other objects--it doesn't need to display anything. We want the DrawGUI object to be created when the Applet object is created--which is why we placed that code in the init() method.

Those of you who are very observant probably notice that this is a new line of code...

```
setVisible(false);
```

It's necessary to set make instance of the GradesApplet object, displayed in the Browser, invisible. In the previous version of the program we executed the exit() method of the System object--but that's not necessary here.

Let's run the GradesCalculation Project in a Web Browser now

That would be great, but unfortunately, our project uses Swing Components, and as I mentioned earlier, that will complicate things a little bit.

First of all, current versions of Internet Explorer and Netscape don't know what to do with Swing Components. In order to run an Applet containing Swing Components in either of those two browsers, it's necessary to first go out to the Sun Website and download what is known as a Java Plug-in. You can read more about it here

<http://java.sun.com/docs/books/tutorial/uiswing/start/swingApplet.html>

and download it from here

<http://java.sun.com/products/plugin/>

Once you have the plugin downloaded and installed (don't worry, it's painless), you might be inclined to code up an HTML file the way we did a few minutes ago and expect the Grades Calculation project to be displayed in your favorite Web Browser.

We're close--very close to that---but the Plug-in will complicate matters. Instead of an HTML document

that looks like this

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

instead it needs to look like this...

```
<HTML>  
<BODY>  
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" WIDTH = "1200" HEIGHT  
= "1200"  
codebase="http://java.sun.com/products/plugin/1.3/jinstall-13-win32.cab#Version=1,3,0,0">  
<PARAM NAME = CODE VALUE = "GradesApplet.class" >  
<PARAM NAME="type" VALUE="application/x-java-applet;version=1.3">  
<PARAM NAME="scriptable" VALUE="false">  
</OBJECT>  
</BODY>  
</HTML>
```

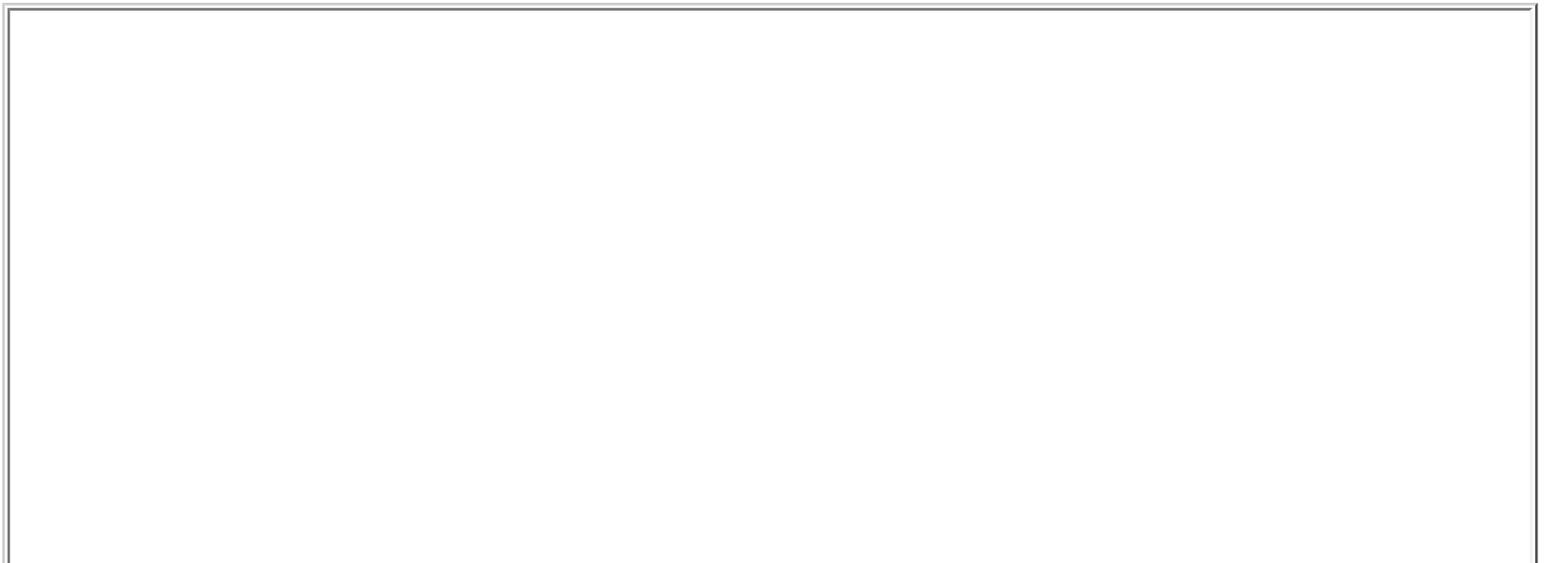
Looks complicated, doesn't it? No need to concern yourself with it too much--just two lines of code require our attention. First, I've expanded the size of the window within the browser via this line of code

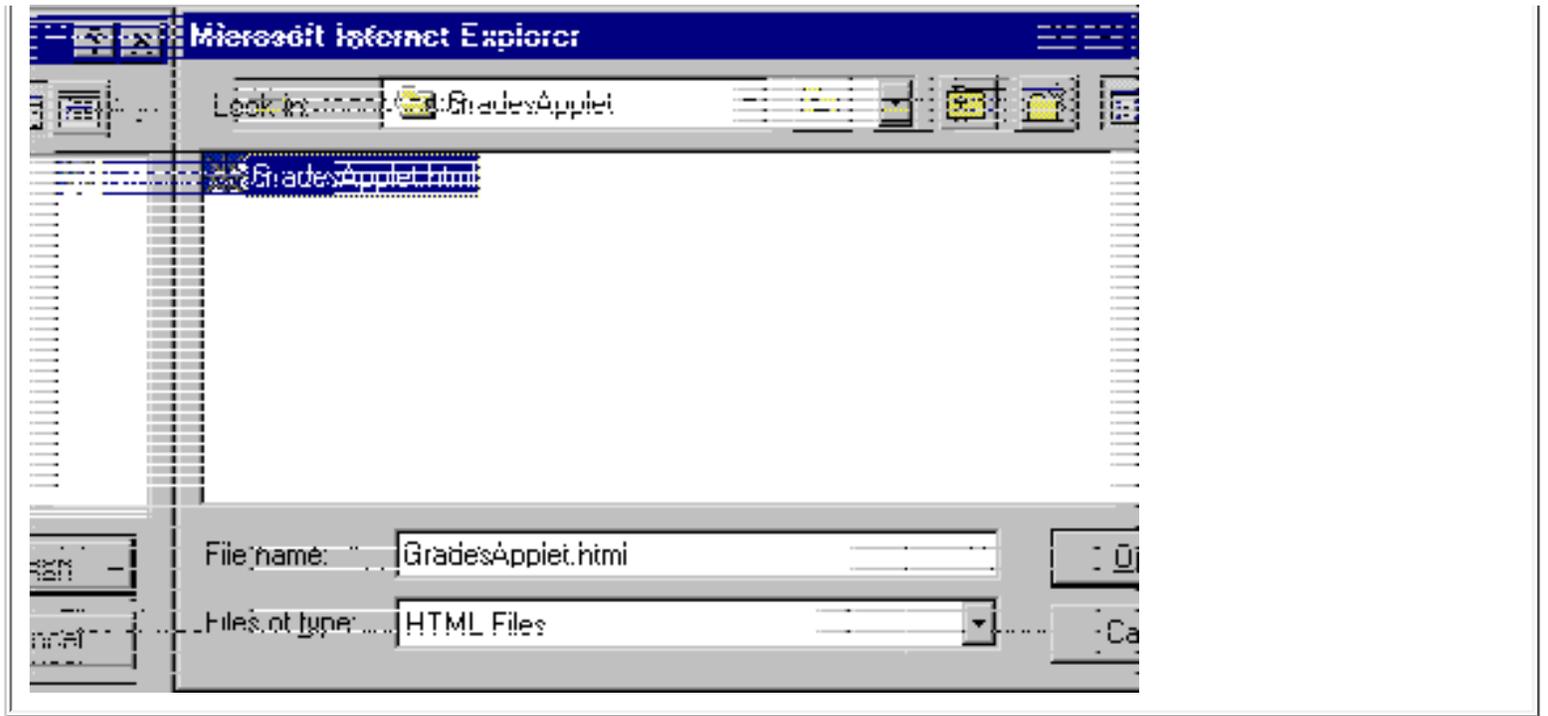
```
WIDTH = "1200" HEIGHT = "1200"
```

and it's this line of code that tells our browser that we wish to instantiate an instance of or GradesApplet class...

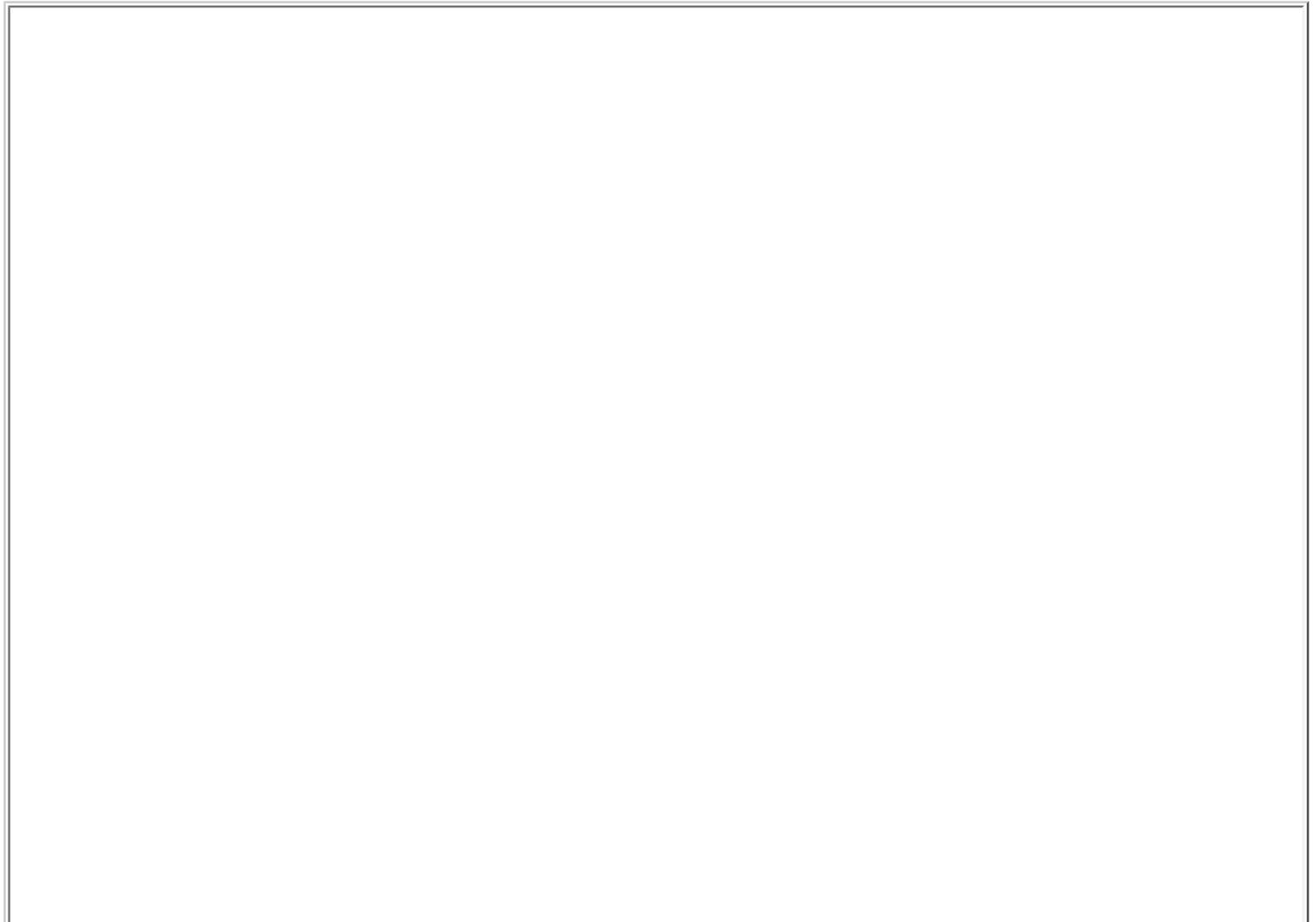
```
<PARAM NAME = CODE VALUE = "GradesApplet.class" >
```

Having compiled GradesApplet into a ByteCode file (in a directory called \JFiles\GradesApplet) I can then open this page in a Browser





and magically, there's our Grades Calculation Project running in a browser.



Student Types	Grades	ENGLISH STUDENT
<input checked="" type="radio"/> English Student	Midterm	<input type="text"/>
<input type="radio"/> Math Student	Final Exam	<input type="text"/>
<input type="radio"/> Science Student	Research Paper	<input type="text"/>
	Presentation	<input type="text"/>

Calculate Grade

Reset

Java Applet Window

Except for the fact that the window says 'Java Applet Window' at the bottom, we wouldn't even know that the project is running in a browser.

Test the program--you should find that the functionality is identical to that of the other version---it should be, all we've done is change the Startup class to be an Applet.

The screenshot shows a Java Applet window titled "Grade Calculator". The window has a blue title bar with standard window controls (minimize, maximize, close). The main content area is light gray and contains the following elements:

- Student Types:** Three radio buttons are listed: "English Student" (selected), "Math Student", and "Science Student".
- Grades:** Four text input fields are arranged vertically, each with a label to its left:
 - "Midterm" with a value of 70
 - "Final Exam" with a value of 80
 - "Research Paper" with a value of 90
 - "Presentation" with a value of 100
- Buttons:** Two buttons are located on the left side: "Calculate Grade" and "Reset".
- Output:** On the right side, the text "Numeric Grade: 84.5 Letter Grade: C" is displayed.
- Footer:** The text "Java Applet Window" is visible in the bottom-left corner of the window.

Now just close the window--and the Applet stops.

Summary

I hope you've enjoyed this article on creating Java Applets. I hope it's given you the confidence to create Applets of your own.

Those of you who purchased my Java book, many thanks!