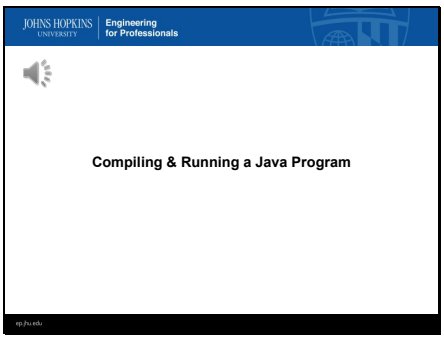
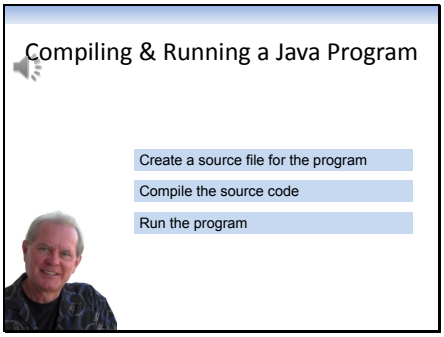
1



In this lecture you will learn how to create, compile and run a Java program.

2

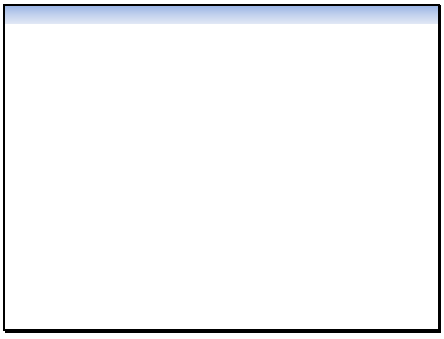


The first step in program development process is to create a source file for your Java program. A source file is simply a file that the program's source code is stored in.

Then, you can compile it into bytecode, and if the program compiles successfully you can run it using the Java interpreter.

I'll illustrate each of these steps for you.

3



To create a source file, you'll need an editing program like a word processor. In this lecture, I'm going to use the Windows Notepad Editor to do this.

I'm going to create, compile, and run a program called JavaDemo. Using Notepad, I create a new document.

Then, I'll type the class structure that will contain the code for this program.

```
public class Java Demo
{
}
```

Next, I'll add a main() method that will contain the programming instructions.

```
public static void main( String [] args )
{
```

}

Then, I'll type in some programming instructions.

System.out.println("Today is a great day for Java programming");
System.out.println("It most certainly is!!");

This program will output two strings to my computer's video display when I run it.

You may have noticed that when I created the class for the program I typed the matching set of braces first, and when I typed in the main() method, I typed in its matching set of braces before I typed any programming instructions. You don't have to do it this way, but sometimes it is hard to remember to add the closing braces, so doing it this way helps me avoid problems.

Now I have my program source code all ready, and I need to save it to my file system. In the last lecture, I created a directory, or folder, on my file system called MyJavaPrograms that I'll use to keep my programs in.

There are 3 important things to remember when saving a source file. First, the name of your file must match the name of your program exactly. Second, the file must have a .java extension. The third thing to remember is that the file must be saved as a text file.

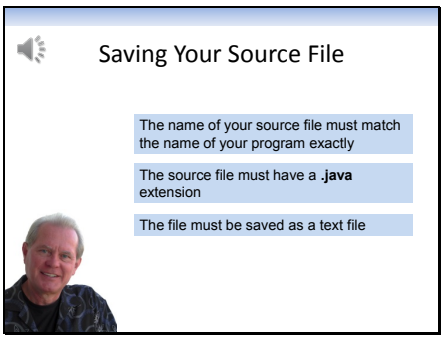
So, I need to name this file JavaDemo.java.

Windows Notepad saves files as text files and uses a default file extension of .txt, so I'll need to override the default extension and use .java instead.

If I was using a word processing program, like Microsoft Word, I'd have to remember to save the file as a text file instead of Word's normal format.

I'm now ready to move on to the next step.

4



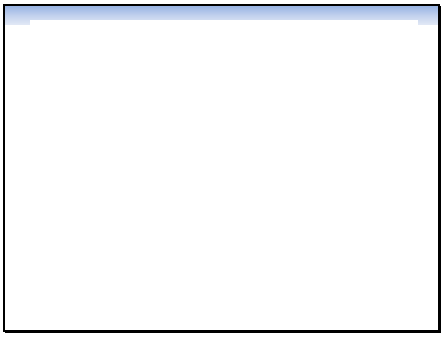
There are three things you must remember when saving your source file:

First…the name of your source file must match the name of your program exactly.

Second…the source file must have a .java extension.

And third…the file must be saved as a text file.

5



I'm now going to compile my program from the Windows command line.

I'll click on Start and then on the command prompt icon which happens to be on my Start menu. If the command prompt icon wasn't on my Start menu, I'd have to click on Accessories and then on the command prompt icon. In this case, I'll just click on the icon from the Start menu. When I do this a Command Shell Window will open.

Next, I need to make my current directory the directory where my Java programs are stored, so I'll type cd followed by the directory name. Now, I'm in the right place.

The name of the Java compiler program is 'javac', and to compile my program I just need to type javac followed by the name of the program. It's important to include the .java file extension. All that remains is to press the enter key. Assuming my program doesn't contain any errors I should see the command line prompt in a few seconds. Let's give it a try.

Good. There were no programming errors. If there were any errors in the program they would have been displayed on the screen, and I'd have to go back and modify the source file to remove the errors.

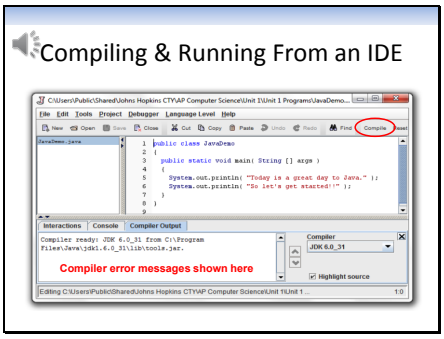
When I compiled this program the Java compiler created a new file on my file system named JavaDemo.class. This file contains the special byte-code that the interpreter will execute. Let's check to make sure.

Now I have my Java source file, JavaDemo.java, and my compiled bytecode file, JavaDemo.class, in my directory.

At this point our program is compiled and ready to be run by the Java interpreter. To run the program I simply type java JavaDemo, hit enter, and voila…the program displays its output for us.

When you run the program through the interpreter you just specify the name of the program. You don't add the file extension .class. The interpreter knows to look for a file with the .class extension. If you typed the .class extension you would get an error message.

6



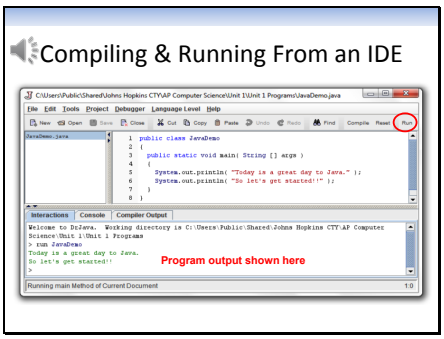
You can also create, compile, and run your Java programs from an Interactive Development Environment or IDE.

In this example, I'm using the DrJava IDE. I typed my program source code right into the tool's editor window.

Note the color syntax highlighting. The syntax highlighting feature can be very useful when visually inspecting your code for errors.

To compile the code, I simply select the Compile option on the toolbar .

Any compiler error messages will be displayed in the Compiler Output window.



Once the program is compiled, I can run it by selecting the Run option on the toolbar.

And the program output will be displayed in the Interactions window, as illustrated here.