Introducing Java, Command Line, and NetBeans

Contents

Introducing Java, Command Line, and NetBeans	1
Overview	2
Java	2
Virtual Machine	2
Two Types of Executable	3
NetBeans	3
The NetBeans Site	4
Downloading	5
Java JRE and JDK	5
Packages and Classes	6
Methods	6
Properties	7
The Math Class – An Example	7
Drilling Down to Details	9
Preliminary Activities	9
Writing a Java Program	10
DOS	12
Easy Use of DOS	14
DOS Summary	17
Compiling a Java Program	18
Path Problems	19
DOS Arrow Use	21
About the Class File	21
Changing your File	21
Altering the Program	22
Applets	24
Creating an Applet	25
Writing an HTML File for an Applet	26
HTML Tags for Applets	27
Using the Applet Viewer	27
Open the Web Page	28
Jar Files	30
Using NetBeans	30
Starting a Project	30
Add a Java File	32
Setting the Main Class	34
Compiling an Individual File	35
Note on the File and Package	36
Debugging	37
Items in the Tasks Table	38
NetBeans and JApplet	39
The HTML for the JApplet	43
Editing the HTML File.	44
Finishing the Poem	45
Viewing the HTML	46
Opening and Closing Projects	47
Other Topics to Review.	48

Overview



This lab introduces you to the essentials of Java development and how to set up a development environment for Java. Java was originally developed to be platform independent, so as much as possible was done to ensure that anyone who has access to the command prompt of a computer and the most elementary text editor can create Java applications. In this lab, only the Windows operating system (DOS) is considered.

Java

Java was developed during the early 1990's by Sun Microsystems. Its key feature was that it was "platform independent." This means that any program written in Java will compile on any computer system. This was a fairly significant feature for a programming language during the early 1990's since at that time, when programmers wrote a program that would compile on a given type of operating system , they would have to rewrite it to make it run on another operating system. Despite the appearance that there are only a few operating systems, the fact is that there are many thousands of different systems used for different things. Most people might think of a PC, a Mac, or a database or internet server, but there are also a multitude of systems that perform such actions as controlling a refrigerator, a toaster, an oven, a car radio, an interface for a hand-held control device for heavy equipment, a phone, or a toy.

Virtual Machine

The developers at Sun were considering this market, and they came up with the idea of a way to develop a programming language that would compile, not to an operating system, but to a *virtual machine*. A virtual machine is a program that runs programs. Sun would make virtual machines for any operating system on which anyone wanted to run a Java program. This might sound involved, but consider that a virtual machine can be developed by a programmer in very little time, This single program can be used to support many thousands of programs.

Two Types of Executable

Twenty years later, Java has become one of the world's leading computer programming languages, especially with respect to applications developed for the internet. In this respect, Java offers two main forms of executable program. One is called a *console program*. Such a program executes on your computer without the use of a browser. The other is called an *applet*. An applet is a program that executes within a web browser. To a great extent, the capacity of Java to run on any machine made it an ideal language for development of applications that could run in browsers.



In this lab, we work with both console applications and applets.

NetBeans

NetBeans is an integrated development environment s (IDE) for Java developers. It runs on Linux and Windows systems. It has been developed with Java. The history of NetBeans goes back almost as far as Java. It was originally a student project, started by some students in the Czech Republic. It was an imitation of one of the first effectively designed Windows IDEs, a program developed by Boreland called Delphi. In 1999, Sun took over NetBeans, and from that point forward it continued to evolve.

NetBeans is far from the only Java IDE. There are hundreds of development environments that have been introduced over the years for Java. One is ConTEXT. Another is Eclipse. One that was very popular for a time is JBuilder, which is now available only in a commercial edition. Most Java IDEs are freeware or open source.

Since Sun (now Oracle) decided to endorse NetBeans as its IDE of choice, it has become the premier IDE for Java development. The only closely competing IDE is Eclipse, which is popular in academic settings. Eclipse became popular because it supports many programming languages. This now becoming true for NetBeans, which each year support more languages and tools. In addition to allowing you to work with Java, NetBeans currently provides modules that support C++, Python, Perl, and PHP. In addition, it supports JRun for testing and provides an HTML editor and other tools. Its full use far exceeds the scope of this course.

The NetBeans Site

1. For exploratory purposes, go to the NetBeans site.

http://netbeans.org/

2. Inspect the website. Here is the current appearance of the NetBeans website.



- 3. As you can see under the **Community** tab, as on open-source system, NetBeans is maintained by an organization of people who work on a voluntary basis to maintain it. They make the software available for general use. In some cases, corporations might pay developers to work in an indirect way to work on NetBeans. This arises because a corporation can derive benefits from the software by indirectly investing in it.
- 4. The **Plugins** tab allows you to see modules that you can download and add to your installation of NetBeans.
- 5. The Docs & Support tab provides endless information on how to use NetBeans to develop with Java. It also addresses other languages, such as C++ and PHP. What is available depends on what different developers have made available. In addition, under this tab you find sample applications.
- 6. The **IDE** tab provides documentation and examples on the main capabilities of NetBeans.

Downloading

- 1. From the NetBeans.org site you can go to download the current version of NetBeans. Note that the version is likely to change once or twice each year. When you upgrade to the newer versions, NetBeans makes it so that all your projects can be transferred automatically from the previous version, and you can maintain several different versions on any one computer at a given time.
- 2. Note that in the context of the classroom, the ITS group does the installation work for us. When you are working on your own, you will need to install the IDE for yourself. Downloading and installing the NetBeans IDE is wholly automated for Windows other supported systems (Linux and Mac). Generally, when you install it, first download the installation executable. Then perform the installation from there.
- 3. You can also download NetBeans from the Oracle site. In this situation, it is offered a bundle with Java. One advantage of downloading the bundle is that you get both NetBeans and the Java support programs, and both are installed at the same time by the installation program.
- 4. If you obtain the two software packages separately, you must first install the software from Oracle that supports Java before you can install NetBeans. If you try to do otherwise, the NetBeans installation package advise you about what you must do first. Generally, the software you require from Oracle is called the Java Development Kit (JDK).

Java JRE and JDK

Java was developed in the early 1990's. The development team was headed by James Gosling, a legendary figure in programming history.

- 1. One of the key characteristics of Java is that it is an object-oriented language (which was far more important in the early 1990s than it is now). As an object-oriented language, it requires that you write your programs as classes.
- 2. Another key feature of Java, as has already been mentioned, is that is uses a virtual machine (VM). A virtual machine is a program that runs programs. Any computer that is to run Java programs much must have a VM installed. The VM is known generally as the **Java Runtime Environment (JRE)**. This program is available free of charge. To obtain it, you simply download and install it.
- 3. To *develop* Java programs, you require a **Java Development Kit** (**JDK**). The JDK consists of over 4000 class definitions organized into hundreds of packages that you can use as you develop Java applications. As with the JRE, you download the JDK from the Oracle site.
- 4. To familiarize yourself with the scope of the JDK, see the following website.

http://download-llnw.oracle.com/javase/6/docs/api/

5. See the following figure. Note that if you intend to work with Java to any extent whatsoever on an academic, independent, or professional basis, you should bookmark this site (or the site corresponding to any version of the JDK you are using). It is impractical to buy books on the class library because they cannot be updated quickly enough to accommodate releases of the software. However, Sun has produced very useful summaries of the libraries over the years, and presumably, Oracle will continue to do the same.



Packages and Classes

If you look at the Java Platform Standard Edition website, you see that the information is broken up into two general groupings, **classes** and **packages**. The terms prove essential to understanding how to use Java.

- 1. A package is more or less analogous to a file folder or directory on a computer. A package usually contains one or more classes.
- 2. A class is an abstract data type. It consists of methods and properties.

Methods

A *method* is another word for a *function* or *subroutine*. Some classes have many methods, others only a few. To use a method, it is important to know how to use its parameters.

Some methods are static, which means that they are called using the name of the class in which they are defined.

Given that there are roughly 4000 classes in the java class library, there are many more thousands of methods. Some classes, such as Math, contain fifty or more methods. Others contain only a few.

Properties

Some Java classes provide properties. Another name for a property is *attribute*. Still another name for a property is *field*. A property is an element of a class to which you can assign values.

The Math Class – An Example

- 1. To get a sense of how to use the library site and what a class and its methods involve, scroll down the class list until you reach the Math class.
- 2. Click the Math class. The following figure shows the top of the class page.



Note. The development team for Java was one of the first programming teams to begin documenting class libraries in this way. Now the practice is universal.

3. Inspect the page. At the top of the page for the class, you see the package in which the class is to be found (java.lang).

- 4. Next inspect the information on the *hierarchy* of the class library. (This topic will be covered in detail later.) The position of a class in a hierarchy determines what it *inherits* from parent classes and what is inherited from it. The Math class inherits certain capabilities from the Object lass. The Object class is the most primary class in Java. As with ActionScript, almost all classes in Java are *descendents* of (or *derived from*, or *children of*) the Object class.
- 5. Scroll down the Math class page. You see two sections: Field Summary and Method Summary.



- 6. Note the following:
 - A *field* is a value that the class contains that can be accessed in much the same way that a method can be accessed. Such values are sometimes referred to as *properties*. The values **E** (natural log) and **PI** are properties of the **Math** class. They are values, in other words, that you can automatically use any time you use the **Math** class.
 - The list of methods follows. Each method is identified in terms of its *return values* and its *parameters*. The return value is a value that the method delivers to any place in a program in which you *call* the method. A parameter is a value that you *pass* to a method. Many methods require several parameter values, and some do not require any parameter values at all.
- 7. Note that after you become familiar with the control features of a language and how to use basic algorithms, the major task you face as a programmer working in the capacity of an application developer is learning how to use all the methods provided by the class library.

Drilling Down to Details

1. Find the method that reads as follows:

```
static double max(double a, double b)
```

Note: The name of the method is max(). The parameters are identified as a and b. The data types of these parameters are of the data type double. *Static* means that the method is called using the name of the class (Math.max(5.0, 7.65), for example.) The word double preceding the name of the method is the type of the returned value (also expressed as *return type*).

- 2. Click on the link (the name of the method).
- 3. You then see the explication of the max () method.

4. Generally, all method descriptions are similar. A body of text explains what the method does. The parameter values are explained, and the values returned, if any, are also explained. Attention is given to what the method does and how it is used. If you read the text of this explanation, you can see that what this method does is fairly simple. It takes two numbers as parameters (arguments) and returns the value of the larger.

Preliminary Activities

Because Java was designed to make it so that you can use it without dependencies on IDEs, this section deals with developing programs using a simple editor, the Command window of DOS. In this widow, you issued DOS commands at the command line.

To compile and run a Java program, you must first install the JDK and the JRE. Oracle makes it very easy to install the JRE since the JRE is intended for use by general computer users rather than programmers. It is analogous to the Flash player, and its primary use is to support Java applets embedded in web pages or applications that run on the desktop. (An *applet* is a program that runs in a web page. An *application* is one that runes on the desktop.)

To be able to compile programs in Windows environments, you must issue a command or set the Path variables so that the computer will be able to recognize the commands you issue. If you want the story from the ground up, here is the Oracle site:

http://download.oracle.com/javase/tutorial/getStarted/cupojava/win32.html

Writing a Java Program

Note: If your computer in the lab does not allow you to access the command line, then this portion of the lab cannot be performed. Even if you cannot perform the tasks given in this portion of the lab, however, it is a good idea to familiarize yourself with it.

- 1. Create a folder on your desktop and name it Java0100.
- 2. Access the Notepad application in the Program list. (Press Window r and type Notepad.)



3. Highlight and copy the following code into Notepad.

```
public class HelloWorld {
    /**
     * Prints Hello World! to the prompt
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
```

}

Note that the file defines a class named Helloworld. The class contains one method, main (). All classes that are to run as applications must contain a main () method that is named and defined as you see in the code example. The code that the main () method contains can vary endlessly, of course. The only

action that this program performs is to print out the expression "Hello World" to the command line. The line of code that makes this happen is the following:

```
System.out.println("Hello World!");
```

Note that the lines that beginning with /** and ending with */ are comments. These lines are not read by the compiler. This is a form of comment that is used for a special form of documentation called Javadocs.

- 4. Save the file as Helloworld. java to the Java0100 directory on your desktop.
 - a. The Encoding must be set to ANSI.
 - b. Type the file type, java, following a dot (or period) after the name of the file. See the following figure.

Save As							? 🔀
Savejn:	🚞 Java0100			~	G 🦻	ø	
My Recent Documents							
Desktop							
My Documents							
My Computer							
	File <u>n</u> ame:	HelloWorld.java				~	<u>S</u> ave
My Network	Save as type:	Text Documents	(*.txt)			~	Cancel
	<u>E</u> ncoding:	ANSI				~	

5. You then see the file in Windows Explorer. You can now open it in either Notepad or Wordpad. Each time you save it, make certain you select the same save options.

Construction (safety (same))	4100			
🔾 🗢 📕 🕨 Java0100		- € 9 Se	earch Java0100	Q
<u>File Edit View T</u> ools <u>H</u> elp				
Organize 🔻 📄 Open S	hare with 🔻 🛛 Burn 🔹 New folder		:==	• 🔟 🔞
☆ Favorites	Name	Date modified	Туре	Size
🐌 Downloads				
 Recent Places Desktop CU - Shortcut 	🗋 HelloWorld.java	2/15/2011 7:59 PM	JAVA File	1 KB
MSSE600 Shortcut				
HelloWorld.class Dat CLASS File	te modified: 2/15/2011 8:45 PM D Size: 426 bytes	ate created: 2/15/2011 8:45 PM		

DOS

Disk Operating System (DOS) is the operating system used for Windows. Although, technically, Windows itself is an operating system, it remains that behind the scenes, DOS is at work. DOS is a command-line system, as are all other operating systems.

1. Press the Window key and r. In the dialog that appears, type **cmd** and click OK. See the following figure.

Run	? 🔀
-	Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:	cmd 💌
	OK Cancel Browse

2. A DOS command window appears.



3. Set the color of the window by selecting Properties from the control icon on the upper left of the command window. Click the Colors tab. (See the following figure.)

"C:\WINDOWS\system32\cr	nd.exe"Properties 🛛 🕐 🔀
Options Font Layout Colo	rs
C Screen <u>I</u> ext Screen <u>Background</u> <u>Popup Text</u> Pop <u>up Background</u>	Selected Color Values <u>B</u> ed: 255 ÷ <u>G</u> reen: 255 ÷ Bjue: 255 ÷
Selected Screen Colors	
C:\WINDOWS> dir SYSTEM (DIR) SYSTEM32 (DIR) DEODME TYT 20	10-01-99 5:01 10-01-99 5:01 10-01-99 5:01
Selected Popup Colors	
C:\WINDOWS> dir SYSTEM <dir> SYSTEM32 <dir> DEGIME TVT 22</dir></dir>	10-01-99 5:01 10-01-99 5:01 10-01-99 5:01 10-02 1:0-01 с-01
	OK Cancel

- 4. Click the radio buttons and the color picker selections to set the Screen Background to white and the Screen Text to black. Click OK when you finish.
- 5. Click OK to save the properties for future windows with the same title.

nt window only
e windows with same title
Cancel

6. What you see varies according to the configuration of your computer, but the command window appears as follows after you have reset the font and background colors. (See the following figure.)



Easy Use of DOS

1. A programming language referred to as DOS (like the operating system) allows you to interact with DOS. In this context, it is not our concern to learn a great deal about DOS, but you can see the basic commands if you type the following at the command prompt:

help

Note. Generally, if you write programs that work at the DOS level, they are called *batch* programs. The programming language used in such programs is called QBASIC (which can be viewed as an extension of DOS). Another term for such a program is *shell script*. A DOS shell script is a batch program. Usually, programmers write scripts to automate the execution of the actions we perform in this section.

2. After you type help at the command prompt and press return, you see the full range of commands available to you. (See the following figure.)

📾 C:\WINDOWS\system32\cmd.exe	- 🗆 🗙
Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp.	_
C:\Documents and Settings\HP_Administrator>DOS help 'DOS' is not recognized as an internal or external command, operable program or batch file.	
C:\Documents and Settings\HP_Administrator>help For more information on a specific command, type HELP command-name ASSOC Displays or modifies file extension associations. AT Schedules commands and programs to run on a computer. ATTRIB Displays or changes file attributes. BREAK Sets or clears extended CTRL+C checking. CACLS Displays or modifies access control lists (ACLs) of files. CALL Calls one batch program from another. CD Displays the name of or changes the current directory. CHCP Displays or sets the active code page number. CHDIR Displays or modifies the checking of disk at boot time. CLS Clears the screen. CHKNTFS Displays or modifies the checking of disk at boot time. CLS Clears the screen. CMD Starts a new instance of the Windows command interpreter. COLOR Sets the default console foreground and background colors. COMPACT Displays or alters the compression of files on NIFS partitions CONVERT Converts FAT volumes to NIFS. You cannot convert the current drive. COPY Copies one or more files to another location. DATE Displays or sets the date. DEL Dielets one or more files.	S-
4	• //

3. At the command line, type the following command and a space. Do not press Enter; just type the two letters and a space following:

cd

4. Your window appears as follows:



5. Now open the directory you just created on your desktop (Java0100).

C: Wocuments and Settings\HP_Administrator\Desktop\Java0100	🛛
<u>Eile E</u> dit <u>Vi</u> ew Favorites <u>T</u> ools <u>H</u> elp	
Address 🗁 C:\Documents and Settings\HP_Administrator\Desktop\Java0100	💌 🄁 Go
🕜 Back 👻 🔊 👻 🏂 Search 🔊 Folders	
Folders × June HelloWorld.java	
🗉 🛅 CU Archive 💽 🚺 Java Language Source file	
🗀 Java0100 🔚 🔚 🕬	
🗷 🧰 Resumes 💽	

6. In the Windows Explorer window, click and hold on the on the name of file you created (Helloworld.java) and drag it over to the DOS command window. Release it. The path is automatically copied to the DOS window:



7. To be able to reach the correct directory, delete the name of the file from the characters that have been copied. To accomplish this, working in the command window, carefully use the left arrow key to move the cursor to the final 'a' of "java" and then press the Delete key once. Then press the Backspace key and delete all the characters up to the last "0" in the name of the directory. See the following figure.



8. **Press Enter.** The CD command executes and you see that the path to the directory you created on your desktop is now showing.



- 9. Type the following command at the command prompt and press Enter: dir
- 10. You see the contents of the directory, which for now holds only your HelloWorld.java file. (See the following figure.)

📧 C:\WINDOWS\system32\cmd.exe	- 🗆 🗙
C:\Documents and Settings\HP_Administrator\Desktop\Java0100>dir Volume in drive C is HP_PAUILION Volume Serial Number is 73BC-4457	-
Directory of C:\Documents and Settings\HP_Administrator\Desktop\Java0100	_
08/28/2010 12:01 PM 〈DIR〉 08/28/2010 12:01 PM 〈DIR〉 08/28/2010 12:01 PM 〈DIR〉 08/28/2010 12:04 PM 〈JS6 HellowWorld.java 1 File(s〉 〈456 bytes 2 Dir(s〉 141,759,791,104 bytes free	
C:\Documents and Settings\HP_Administrator\Desktop\Java0100>_	_
•	

DOS Summary

The following table provides a summary of the most frequently used DOS commands. Note that you can type these commands as either capital or lower-case letters. To obtain specific information on any one command, type the word **help** followed by the command. For example, for help on **dir**, type the following command:

help dir

*	This is an asterisk, It is called a wildcard. You can use it to replace any character. For example, to check a directory for HelloWorld.java, you can type DIR H*. The asterisk tells the system to look for any file name beginning with H. You might also type *.java to see all the files that are of the type java.
CD	This changes the current directory. When you follow this command with two dots (periods) you navigate up a level (<u>CD</u>) To navigate up two or more levels, use dots in combination with slashes (<u>CD</u> \ or <u>CD</u> /). If you want to navigate down a directory, then you follow CD with the name of the next level down. You can use the slashes to navigated down several directories, separating them using the slash. (CD/nextDown/nextDownAfter)
CLS	Clears the screen.
COPY	Copies one or more files to another location.
cmd	Starts a new instance of the Windows command interpreter.
DATE	Displays or sets the date.
DEL	Deletes one or more files. Use this for files. Type <u>DEL file_name</u> . For the HelloWorld.class file, for example, you type del HelloWorld.class .
DIR	Displays a list of files and subdirectories in a directory. A significant addition to the DIR command is DIR/P, which allows you to see the contents of a directory a page at a time.
ECHO	Displays messages, or turns command echoing on or off.
EXIT	Quits the CMD.EXE program (command interpreter).
HELP	Provides Help information for Windows commands.
MD	Creates a directory. Same as MKDIR.
MKDIR	Creates a directory. Type the command and then the name of the directory you want to create. <u>MKDIR <i>directory_name</i></u> .

RD	Removes a directory. Removes a directory. Use with caution. To remove a directory, remove its contents first. The system prompts you to ensure that you want to delete the contents. You type <u>RD directory_name</u> .
REN	Renames a file or files. You type the command, the name to be changed and the new name: REN old name new name
RENAME	Renames a file or files. (Same as REN.)
TIME	Displays or sets the system time.
TITLE	Sets the window title for a CMD.EXE session.
TREE	Graphically displays the directory structure of a drive or path.
TYPE	Displays the contents of a text file.
VER	Displays the Windows version.
VOL	Displays a disk volume label and serial number.
XCOPY	Copies files and directory trees.

Compiling a Java Program

1. To verify that you can perform the actions detailed in this section, issue the following command at the command prompt:

java -version

2. The response to this command is that you see the current version of the Java JDK. If you see no response of this type, then you must use a special approach to compiling your program. Jump momentarily to the next section if this is necessary.

```
C:\Users\john\Desktop\Java0100>java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) 64-Bit Server VM (build 19.0-b09, mixed mode)
C:\Users\john\Desktop\Java0100>
```

3. To compile your Java file, type the following command at the command prompt and press return.

```
javac HelloWorld.java
```

4. After you issue the command, if your program contains no errors, and the Java compiler has been configured correctly, you see only that the prompt is refreshed. This signals success.



5. Issue the following command at the command prompt to see the newly created *.class file:

dir



6. Now run you new Helloworld program. To accomplish this task, issue the following command at the command prompt.

java HelloWorld

7. The program executes, and you see the message that it delivers printed at the command prompt.



Path Problems

A "path" is a directory path stored in your computer that allows it to know the location of a given program. If the path is set for the program, then you can execute the program for any directory on your computer. If it is not set, then you must explicitly state the path before you can execute the program.

Refer to this section if you are having problems with getting the javac, java -version, or other Java commands to work. It is likely that the class path is not set. As mentioned

previously, the class path tells the computer where to find the executable that compiles your Java file.

Normally, if the path is set, to compile a java program, you open a DOS command window and issue the following command:

javac HelloWorld.java

If the path is not set, however, you run into problems because the system cannot respond to the command. You see a message that tells you that the program cannot be compiled. The system cannot find the javac program—which is the compiler for Java.

To overcome this difficulty, use the following approach:

- 1. Follow the instructions given earlier and have a DOS window open.
- 2. To find out which path to use, use windows Explorer to navigate to the Java directory in which the Java JDK has been installed. It is likely to be in the **Program Files** directory. Navigate to this directory and then find the **Java** subdirectory. From there navigate to the **bin** directory. In the bin directory, you find the java executable. See the following figure.



3. For the path shown in the previous figure, then, you can copy out the information you need and prefix it to the javac command. The command with the qualifying path appears as follows (yours might differ): You must enclose directory names that contain spaces with double quotes (as with "Program Files").

C:\"Program Files"\Java\jdk1.6.0_01\bin\javac HelloWorld.java

Here is exactly how the command for the previous section (given the computer being used) is issued. The command prompt is shown in colored italics. This is the path to the directory in which Java file resides. The path to the javac program is shown in dark type.

C:\Documents and Settings\HP_Administrator\Desktop\Java0100>C:\"Program Files"\Java\jdk1.6.0_01\bin\javac HelloWorld.java

4. The command you execute is issued from the prompt of the command window, and when you issue it, you must have navigated in the command window to the same directory that contains the file you want to execute.

DOS Arrow Use

Note that you can use the up and down arrow keys in DOS to repeatedly issue commands you have already issued. This makes it to that you do not have to retype commands. Simply arrow to the command and press enter. Use this option as much as possible.

About the Class File

As you develop a program, you must repeatedly change it and compile it to check whether the changes you have made have resulted in what you are seeking to do.

Each time you want to change a *****.java file and use the javac command to regenerate it, you must first delete the previous version of the *****.class file. The Java compiler cannot overwrite the old *****.class file.

1. To remove an old version of a *****.class file, issue the following command.

del HelloWorld.class

- 2. Issue the dir command to see that the *****.class file has been deleted.
- If you then issue the dir command, you see that the directory no longer contains a *.class file. At this point, reissue the following command to update your work:
 javac HelloWorld.java
- 4. Issue the following command to see the newly generated file file.

dir

Changing your File

1. At the DOS prompt, issue the following command, which duplicates the HelloWorld.java file and saves it under a new name.

copy HelloWorld.java Cocoon.java

2. Issue the following command, which opens the Cocoon.java file with Notepad:

notepad Cocoon.java

3. Change the text of the file so that it reads as follows:

```
public class Cocoon{
    /**
    * Prints Emily Dickison's poem
    * http://www.bartleby.com/113/index2.html
    */
    public static void main(String[] args) {
```

```
System.out.println("From cocoon forth a butterfly");
}
```

4. Compile and execute your file using the procedure detailed in the previous sections. The steps are as follows:

```
javac Cocoon.java
java Cocoon
```

- 5. Recompile and get in a development frame of mind.
 - a. Do not close the Notepad file. Leave it open and use Alt + Tab to switch back and forth between it and the command line.
 - b. At the command prompt, use the arrow keys in the command line to retrieve previous commands and alter them.)
 - c. Here is how you might set up your desktop to work in a convenient way.



- d. After you change your *.java file, save it.
- e. Delete the old *****.class file by issuing the following command:

```
del Cocoon.class
```

Altering the Program

1. Alter the program. Copy the following line as many times as are needed and place a line of Dickinson's poem in each line. The lines of the poem are given in step five.

System.out.println("From cocoon forth a butterfly");

2. Here is an example of how your program appears after the first few lines:

```
public class Cocoon{
    /**
        * Prints Emily Dickison's poem
        * http://www.bartleby.com/113/index2.html
        */
    public static void main(String[] args) {
        System.out.println("\n Emily Dikinson (1830 - 1886) ");
        System.out.println("\n");
        System.out.println(" FROM cocoon forth a butterfly");
        System.out.println(" As lady from her door");
        System.out.println(" Emerged-a summer afternoon-");
        System.out.println(" Repairing everywhere,");
        System.out.println("\n");
    }
}
```

- 3. The complete poem is provided further along (see step 5). Study the code and then finish the poem using the lines provided . Note the following:
 - The \n causes the program to force a line return. You must insert the following line after each stanza:

```
System.out.println("\n");
```

- Space precedes the first letter of each line of the poem, and this is preserved in what you see in the output.
- Add only one stanza at a time. Recompile and execute as you go. Do not attempt to make everything work at once.
- Try working a few lines of the poem at a time, compiling as you go to check for syntax errors.
- 4. Remember as you go that you must delete the *****.class file each time you recompile,
- 5. Here are the lines for the complete poem:

```
FROM cocoon forth a butterfly
As lady from her door
Emerged-a summer afternoon-
Repairing everywhere,
```

```
Without design, that I could trace,
```

Except to stray abroad On miscellaneous enterprise The clovers understood.

Her pretty parasol was seen Contracting in a field Where men made hay, then struggling hard With an opposing cloud,

Where parties, phantom as herself, To Nowhere seemed to go In purposeless circumference, As 't were a tropic show.

And notwithstanding bee that worked, And flower that zealous blew, This audience of idleness Disdained them, from the sky,

Till sundown crept, a steady tide, And men that made the hay, And afternoon, and butterfly, Extinguished in its sea.

Note, the source site for the poem is as follows:

```
http://www.bartleby.com/113/index2.html
```

Applets

When it was first introduced roughly 20 years ago, one of the most noted features of Java was its ability of its programs to run in web browsers. Its ability to do so results from the JRE, which supplements the operations of the major browsers.

When Java applets run inside a browser, they do not directly interact with the operating system of the computer on which they are running. This limitation is usually listed under the rubric of *security*. Does the program have the ability to violate the integrity of the computer on which it is running? The JRE is designed to make it so that this does not easily happen. It provides a secure way of running applications on a computer.

An applet is a program that runs inside a browser, and with few exceptions, it is just the same as a regular java program. It accesses the JRE and runs using it. Since it runs inside a browser, however, it must be embedded in a web page—an html file.

Creating an Applet

- 1. To create a Helloweb applet, begin by once again using Notepad to create a file. This time, call the file Helloweb.java. Save this file to working directory 0100Java.
- 2. Copy the following body of code to the file and once again save it.

```
/*
   HelloWeb.java
*/
import java.awt.Graphics;
import javax.swing.JApplet;
public class HelloWeb extends JApplet {
   public void paint(Graphics g) {
     g.drawString("Hello, World Wide Web!", 10, 50);
   }//end paint
}//end class
```

Note that the text of this program differs from the HelloWorld.java program in fairly significant ways, which will be discussed later on in detail. For now, however, note that the class extends JApplet. JApplet is a class specifically designed to accommodate web pages. The primary method in the applet is the paint() method, and this method attends to displaying whatever the applet displays. To display the message, it uses a Graphics object (named g), which calls the drawString() method.

3. Compile the HelloWeb.java file exactly the same way as you did the HelloWorld.java file. Here is the command you type to compile it:

```
javac HelloWeb.java
```

- 4. After you compile the file, issue the **DIR** command. You see that the **javac** utility generates a ***.class** file.
- 5. Note that this is an applet, so you cannot run it from the command line. To see it execute, you must put it in a web page.

Writing an HTML File for an Applet

To run your applet, you must create an HTML document. The HTML document allows you to perform two types of action. The first is to use a test utility, the *Applet Viewer*, to test your applet. The second allows you to run your applet within your browser.

1. At the command line, issue the following command:

notepad HelloWeb.html

- 2. Save the file to your **0100Java** directory.
- 3. Copy the following HTML code to the file and once again save the file:

```
<html>
<head>
<title>HelloWeb Applet</title>
</head>
<body>
<h1 align=center>HelloWeb Applet</h1>
<center>
<applet name="HelloWeb"
code="HelloWeb.class"
width=150
height=100>
</applet>
</center>
</body>
</html>
```

Note that the *.html extension identifies the document so that your browser can read it. In the HelloWeb.html file, the words in the angle brackets (<>) are HTML tags. The only tag set that you need to include an applet in a web page is <applet</applet>, and you must, at a minimum, assign a value to the code, width, and height properties of the applet tag. This simply names the applet you are trying to run.

The following table provides you with a breakdown of what the tags mean. Opening and closing brackets must enclose each tag, but in the case of the applet tag, you also work with *tag attributes*. Tag attributes fit inside the opening and closing tag brackets. Tags also occur in sets, so for each opening tag, you must provide a closing tag. To create a closing tag, you precede the tag name with a slash (<\>).

HTML Tags for Applets

ltem	Discussion
html	This tag opens and closes an HTML document. Everything you want to include in an HTML document falls between the opening and closing html tags.
head	This tag provides an address space for an HTML document. You use the <u>title</u> tag within the <u>head</u> tag to make it so your browser can display the name of your document.
body	This designates the main part of your web page. Within the body, you place the applet and text.
h1	This tag creates a heading, such as the one you see at the top of this section.
center	This tag causes anything that follows it (text or your applet) to be centered in the web page.
applet	This tag identifies an applet. It has four required attributes: name, code, width, and height. Unless you assign values to these four attributes, your applet tag is not likely to successfully display. The units of measure are picture elements (pixels) of your monitor
name	The name attribute is what you name your applet relative to the web page. You can name your applet anything you want for use within your HTML document.
code	The code attribute is the name of your applet as you have compiled it. Since you have named your file HelloWeb.java and generated a class file named HelloWeb.class, you must assign HelloWeb.class to the code attribute.
width	This is the minimum width of your applet. If you do not make this large enough, your applet might not appear.
height	This is the minimum height of your applet. If you do not designate enough height, your applet might not appear.

Using the Applet Viewer

Before you try to run your applet in your browser, first test it. To test it, you can use the applet viewer. This is a utility program that accompanies the JDK. Its name when you execute it at the command line is **appletviewer**. This utility allows you to test applets without having to invoke a browser. It displays only the output of your applet. To use the applet viewer, you must embed your applet in an HTML page.

1. To test run your applet, type following command at the command prompt:

appletviewer HelloWeb.html

2. You see the following window appear on your desktop.

🛃 Applet V 🗖 🗖 🗙
Applet
Hello, World Wide Web!
Applet started.

- 3. Inspect the viewer. You see only the text that you use the Java code to generate. You see nothing of the HTML page.
- 4. To close the Applet Viewer, click the control (X) button and then select Close.

Open the Web Page

- 1. The applet view is handy, but there are other, practical ways, to test your applet.
 - Open Windows explorer and click on the HelloWeb.html file.
 - Alternatively, type the following command at the command prompt: HelloWeb.html
- 2. You see the following web page:



- 3. Note the feature. The applet in this case consists of the words, "Hello, World Wide Web! The large message at the top is created using HTML.
- 4. **Replace** the code in the HelloWeb.html file with the following:

```
<html>
 <head>
     <title>HelloWeb Applet</title>
 </head>
  <body>
  <h1 align=center>HelloWeb Applet</h1>
  <center>
                 <applet name="HelloWeb"
                      code="HelloWeb.class"
                      width=150
                      height=100>
                 </applet>
              </center>
           </body>
</html>
```

5. Inspect the code. This is a more comprehensive ***.html** file. When you run it, you see the applet inside a table, and in the tab of the browser, you see the name of the file.

🏉 HelloWel	b Applet - Windows Internet Explorer	🛛 🔀
00-	🖉 C:\Documents and Settings\HP_Administrat 💌 🖘 🔀 Google	• •
Eile Edit	<u>V</u> iew F <u>a</u> vorites <u>I</u> ools <u>H</u> elp	
🚖 Favorites	C HelloWeb Applet	
	HelloWeb Applet	

6. At this point, save and close Notepad and the Command window.

Jar Files

There are still a few fundamental operations that it is good to know about when using Java in a command-line context. One central notion is that of a *.jar file. We will examine the *.jar file in another context.

Using NetBeans

The NetBeans IDE helps you develop Java and other files, including *.html files. Tasks become much easier, but at the same time, there is a bit more to understand.

Note. This lab introduces you to uses of NetBeans that fairly restrictive. In future labs, you'll find that NetBeans can be used in a number of flexible ways that free you from having to use either application or applet projects. You can develop any program as an element in a Java class library, so that the project-oriented perspective falls away and you build things from scratch.

Starting a Project

1. Open NetBeans and from the File menu select New Project. The New Project dialog appears.



- 2. Click the Java folder and Java Application. Then click Next. You see the New Project Application dialog.
- 3. As illustrated in the following figure, do the following:

- In the Project Name field, type 0100Java.
- Click the Browse button and navigate to the folder you have created on the desktop. (This folder is is also named 0100Java.)
- Click the two checkboxes at the bottom to **deselect** Create Main Class and Set as Main Project.
- Click Finish.

New Java Application			×
Steps	Name and Loca	tion	
1. Choose Project 2. Name and Location	Project <u>N</u> ame:	0100Java	
	Project Location:	C:\Documents and Settings\HP_Administrator\Desktop\Java0100	Browse
	Project Folder:	ents and Settings\HP_Administrator\Desktop\Java0100\0100Java	
	Use Dedicated	d Folder for Storing Libraries	Descus
	Lipraries Folde	Different users and projects can share the same compilation libraries (see Help for details).	Bro <u>w</u> se
	🗔 Creste Main (Nase HalimAnnid	
	Set as Main F	Project	
	6		
		< Back Next > Einish Cancel	Help

4. NetBeans creates a project file for you. Initially, what you see appears as shown in the following figure.

NetBeans IDE 6.9.1			X
<u>Eile Edit ⊻iew N</u> avigate <u>S</u> ource Ref <u>a</u> ctor <u>R</u> un	<u>D</u> ebug <u>P</u> rofile Tea <u>m</u> <u>T</u> ools <u>W</u> indow <u>H</u> elp		
🔁 🞦 🔛 🧤 (detault contig> 💌	🍸 🦉 🕨 🚯 • 🚯 • 😹 🐂	<u>i</u>) (*	Search (Ctrl+I)
Projects () x : Files	Start Page x		
Source Packages default package> Test Packages		NetBeanside	
<pre> <default package=""></default></pre>	Learn & Discover	My NetBeans	What's Nev
ia - ∰ JDK 1.6 (Default) □ - Can Test Libraries □ - ∰ JUnit 3.8.2 - junit-3.8.2 jar	Recent Projects	Install Plugins	Activate Feature
iain≊a o∪ni 4.3 - juini-4.3 jar	参 JavaExplorationsA 参 0100 JavaLab 📳 AppendixAMIDlets	Add support for other languages and technologies by installing plugins from the NetBeans Update Center.	NetBeans turns on functionality as you us Start creating and ope projects and the IDE \ activate the features \ need, making your experience quicker an
	<	III	>
	;Evaluate Code		₩ ×
	×		> •••
	soutput – oloojava (clean)		₩ ×
(0100 love Main) Nevienter	init:		
; crocsava (mairi) - navigator 🔤 🗴	Updating property file: C:\Docum	ments and Settings\HP_Administrat	or\Desktop\Java0100\0100
	Java\build\built-clean.propertie	-	4
	Deleting directory C:\Documents	and Settings\HP_Administrator\De:	sktop\Java0100\0100Java\
	clean:		
<no available="" view=""></no>	BUILD SUCCESSFUL (total time: 0	seconds)	
			~
Tasks			

Add a Java File

- 1. Click to activate the Files tab in the IDE.
- 2. Click to open the 0100Java tree.



3. Locate the src folder under the 0100Java class folder. Right click on the src folder and select New > Java Main Class. (If you do not see Java Main Class, then select Other. This will show you the Java Main Class option.)



4. You see the New Java Main Class dialog. (See the following figure.)

🕽 New Java Main Class 🛛 🛛 🔀			
Steps	Name and Location		
1. Choose File Type 2. Name and Location	Class Name. HelioWorld		
	Project: 0100Java		
	Location: Source Packages		
	Package: introfiles		
	Created File: ts and Settings\HP_Administrator\Desktop\Java0100\0100Java\src\introfiles\Hello/Vorldjava	я	
		-	
	< Back Next > Einish Cancel Help		

- 5. As illustrated in the preceding figure, in the Class name field, type Helloworld.
- 6. In the Package field, type introfiles. Click Finish.

- 7. You see the main interface of NetBeans once again (see the following figure). Do the following:
 - a. Verify that the Files tab is active.
 - b. Click the src and introfiles folders to drill down to the HelloWorld.java file.
 - c. Click the Helloworld.java file to make it appear in the editing panel. (This should already be the case.)



Setting the Main Class

When working with NetBeans, every project involving an application (rather than an applet) has a main class file. A main class file is a file that contains a main () method. A given project might have several files that contain main () methods (used for testing), but with an application, one file must be the starting point for the execution of the application. This file is the file that is first executed as your project is *built*. Building a project involves assembling all the relevant files into a single application. Not all files in a project need to be part of a single application, however. You can work with many separate files as separate programs that one project happens to contain.

- 1. To designate a main class file, click on the green triangle in the tool bar. (Alternative, press F6.)
- 2. The Run Project dialog appears, as shown in the following figure. Click OK.

🗊 Run P	roject 🔀
(Project 0100Java does not have a main class set.
	Select the main class:
	introfiles HelloWorld
	OK Cancel

3. The project compiles, and the Output panel appears. At this point you see the following message:

```
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Compiling an Individual File

As you add more files to a project, you can compile the files separately by right clicking in the editing panel and selecting Run File. The file you designate is compiled, and you see the output in the Output panel.

- 1. To explore compiling in NetBeans, select and delete all the code in the HelloWorld.java file.
- 2. Copy and paste the following code into the HelloWorld. java file.

```
package introfiles;
```

```
public class HelloWorld {
    /**
    * Prints Hello World! to the prompt
    */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
```

}

- 3. After you paste the code into the panel, select Source > Format.
- 4. Right click either on the file in the src tree on in the editing area and select Run File.
- 5. You see the following output in the Output panel (it appears in the lower part of the IDE, adjacent to the Tasks tab).

```
run:
Hello World!
BUILD SUCCESSFUL (total time: 1 second)
```

Note on the File and Package

This section is optional.

The only change from the previous version of the Helloworld.java file is the inclusion of the package statement:

package introfiles;

Packages are not required, but in this context, NetBeans has created a package for you. Generally, professional programmers will always use packages. The package directive identifies the directory in which the Helloworld.java file is located. This directory is has been automatically generated by Netbeans and is called introfiles. You can confirm this is you glance at the project panel:L



Outside of NetBeans, if you decide to work with the file in the command line context, a few changes in how you run the file must be made. Assume, for example, that you open a command window and navigate to the directory in the NetBeans project folder (src) that contains the HelloWorld.java file.

In this location, you still issue the following command:

javac HelloWorld.java

This command, as you know, generates a *.class file.

In NetBeans, the class file is placed in a colder named classes, which is under the build director. Further it is placed in its package folder, introfiles.

To run the folder, you must cd to the classes folder. Then, in the classes folder, you issue the following command:

java introfiles.HelloWorld

Here is a screen shot showing the issuance of the command:

C:\Windows\system32\cmd.exe	x
C:\Users\john\Desktop\Java0100\0100Java\build\classes>dir Volume in drive C is OS Volume Serial Number is EC51-F62E	•
Directory of C:\Users\john\Desktop\Java0100\0100Java\build\classes	E
02/15/2011 09:43 PM <dir> . 02/15/2011 09:43 PM <dir> . 02/15/2011 09:43 PM 0 .netbeans_automatic_build 02/15/2011 09:44 PM <dir> introfiles 1 File(s) 0 bytes 3 Dir(s) 866,358,636,544 bytes free</dir></dir></dir>	
C:\Users\john\Desktop\Java0100\0100Java\build\classes>java introfiles.HelloWorl	d
Hello World!	
C:\Users\john\Desktop\Java0100\0100Java\build\classes>	

When you issue this command, you are accessing the program through the package that contains it. Any file that is developed using a package must be accessed this way.

When you are working with NetBeans, you do not have to worry about this, however.

Debugging

Debugging is an essential aspect of programming. When working with NetBeans, the Tasks tab is an essential part of debugging. It provide information on syntax and other errors that occur as you are programming.

1. To see how the Tasks tab works, type five dashed in the main function just before the closing brace. As you type the dashes, an exclamation mark appears to the right, showing that you have typed incorrect syntax. that a line of dashes that will cause a compiler error has been highlighted. (See the following figure.)



- 2. Compile. To accomplish this, right click in the edit pane and select Run File.
- 3. You immediately see an error dialog.

🕜 Run P	Project	
<u>^</u>	One or more projects were compiled with errors. Application you are running may end unexpectedly.	
	Always run without asking	
	Run Anyway Cancel	

4. Click Cancel to stop the compiler.

Items in the Tasks Table

- 1. Click the Tasks tab.
- 2. Click the icon on the left to display the tasks:



- 3. Double click on the item that reads "illegal start of expression." Notice where this takes you in the program. When you look at this line, you see that there are no problems.
- 4. Double click on the item that reads "unexpected type." Notice where this takes you in the program. When you look at this line, you see that four expected characters have been typed into the program.
- 5. Delete the four characters (the four dashes).
- 6. From the top menu, select Source > Format.
- 7. Right click in the edit area and select Run File. This time the program successfully compiles, and you see "Hello World!" printed in the Output panel.

NetBeans and JApplet

A Java program that runs in a browser is generally referred to as an applet. When Java was first introduced, the Applet class supported such programs. Over time, however, the developers of Java decided to introduce a revised version of the Applet class. This is the JApplet class. JApplet is derived from Applet and provides a number of improvements. Generally, as you develop applets, derive your classes from JApplet.

Add a new project to NetBeans. To accomplish this, select File > New Project.

1. Select Java and Java Class Library. Then click Next.

New Project		×
Steps	Choose Project	
1. Choose Project 2	Categories: Java JavaFX JavaFX Java Web Java EE Java Card Java ME Maven PHP Ruby CrC++ NetBeans Modules	Projects: Java Application Java Class Library Java Project with Existing Sources Java Free-Form Project
	Description: This feature is not yet enabled. F Creates a new Java SE library library does not contain a main c IDE-generated Ant build script	Press Next to activate it. γ in a standard IDE project. A Java SE lass. Standard projects use an to build, run, and debug your project.
	< Back	Next > Einish Cancel Help

2. You see the New Java Class Library dialog. See the following figure.

🗊 New Java Class Library		×		
Steps		Name and Loca	tion	
1. Choos 2. Name	e Project and Location	Project <u>N</u> ame:	0010JApp	
		Project Location:	C.\Documents and Settings\HP_Administrator\Desktop\Java0100	Browse
		Project Folder:	ents and Settings\HP_Administrator\Desktop\Java0100\0010JApp	
		Use Dedicate	d Folder for Storing Libraries	
		Libraries Folde	er:	Browse
			Different users and projects can share the same compilation libraries (see Help for details).	
			< Back Next > Finish Cancel	Help

- 3. Name the project **0010JApp**. It is not necessary browse to set a new directory. This will be saved to same directory, by default, at the previous project.
- 4. Click Finish.

You see the new project in the NetBeans interface.

5. Click the new project and then the Files tab, so that you see the scr folder.



6. Right click on the scr folder and select New > Java class.



You see the New Java Class dialog.

New Java Class	X		
Steps	Name and Location		
 Choose File Type Name and Location 	Class Name: FirstJApp		
	Project: 0010JApp		
	Location: Source Packages		
	Package:		
	<u>C</u> reated File: C:\Users\john\Desktop\Java0100\0010JApp\src\FirstJApp.java		
	Warning: It is highly recommended that you do NOT place Java classes in the default packa		
	< <u>Back</u> Next > <u>Finish</u> Cancel <u>H</u> elp		

7. As shown in the previous figure, in the New Java Class dialog, name the new class **FirstJApp** and click Finish.

Note that this time around, no package has been created. This is a practice that will be followed in subsequent exercises.

You see the template of a class file in the edit panel.

8. Delete the code of the newly generated file and replace it with the following:

```
import java.awt.Container;
import java.awt.*;
import javax.swing.*;
public class FirstJApp extends JApplet {
    JTextArea outputArea;
   public void init() {
        outputArea = new JTextArea(4, 60);
        Font font = new Font("Verdana", Font.BOLD, 15);
        outputArea.setFont(font);
        outputArea.setForeground(Color.BLUE);
        Container c = getContentPane();
        c.add(outputArea);
        outputArea.setText(
           "Hands by Robinson Jeffers"
           + "\n Inside a cave in a narrow canyon near Tassajara"
           + "\n The vault of rock is painted with hands,"
           + "\n A multitude of hands in the twilight, "
           + "a cloud of men's palms, no more,");
    }
}
```

9. Right click on the FirstJApp.java file in the src tree and select Run File. The Applet Viewer appears, and you see the applet.

Applet Viewer: FirstJApp.class	x
Applet	
Hands by Robinson Jeffers	
Inside a cave in a narrow canyon near Tassajara	
The vault of rock is painted with hands,	
A multitude of hands in the twilight, a cloud of men's palms, no mor	e,
Applet started	
Approcotateou.	

The HTML for the JApplet

At this point, just by running the **FirstJApp**.java file using NetBeans, you have automatically generated class and an HTML files.

1. To view the HTML file, click the **build** and **classes** folders in the Files tab of the project. As illustrated by the following figure, you see the HTML and the class files.



- 2. Right click on the FirstJApp.html file and select View.
- 3. You see the applet displayed in a web page in your default browser.

🙆 Applet HTML Page - Windows Internet Explorer	
🔄 🔄 🖉 C:\Documents and Settings\HP_Administrator\Desktop\Java0100\0010JApp\build\Firs 🗹 🚱 🔀 creating an applet in netbeans	4
File Edit View Favorites Tools Help	
👷 Favorites 🛛 👻 🍘 Use NetBeans to create a ne 🎉 CULink University of Colora 🎉 Applet HTML Page 🛛 🗙	
	_
Applet HTML Page	
	-
Hands by Robinson Jeffers Inside a cave in a narrow canyon near Tas The vault of rock is painted with hands, A multitude of hands in the twilight, a clou	
Generated by NetBeans IDE	~

Editing the HTML File

Notice that the first view you have of the applet you have just created reveals a problem. The problem is that the end of the line is truncated. You see only "Tas" when the poem reads "Tassajara."

This occurs because the size of the applet needs to be changed.

- 1. Double click on the FirstJApp.html file in the Files panel of NetBeans.
- 2. You see the text of the HTML file in the edit window.
- 3. Find the width property an change its assigned value from 350 to 450. Save your work.

You now see the full text.

Applet HTML Page

Hands by Robinson Jeffers Inside a cave in a narrow canyon near Tassajara The vault of rock is painted with hands, A multitude of hands in the twilight, a cloud of men's palms,

Finishing the Poem

}

- 1. Double click on the FirstJApp.java file to reactivate it in the editor window.
- 2. Note the following snippet of code from the FirstJApp.java file.

```
outputArea.setText(
    "Hands by Robinson Jeffers"
    + "\n Inside a cave in a narrow canyon near Tassajara"
    + "\n The vault of rock is painted with hands,"
    + "\n A multitude of hands in the twilight, "
    + "a cloud of men's palms, no more,");
```

- 3. This code differs from what you have worked with before in Java because it consists of concatenated strings.
- 4. The entire string is used as an argument to the **setText()** function.
- 5. To finish the poem, by moving the closing parenthesis of the setText() function down two lines.

```
23
23
23
24
24
25
26
27
26
27
3
28
3
29
```

6. Then on the line above the closing parenthesis, type a plus (+) sign, double quotes, a newline escape sequence (\n), and then the words, followed by closing double quotes. (See the bolded line.) You'll find that NetBeans provides an intelligent text editor, so much of the typing will be finished for you.

```
outputArea.setText(
   "Hands by Robinson Jeffers"
   + "\n Inside a cave in a narrow canyon near Tassajara"
   + "\n The vault of rock is painted with hands,"
   + "\n A multitude of hands in the twilight, "
   + "a cloud of men's palms, no more,"
   + "\nNo other picture. There's no one to say" );
```

7. Note that the line that begins with "a cloud" is not preceded by an escape sequence. This is because the line continues the previous line.

8. Having familiarized yourself with how to add one new line, add the rest of the poem. The text of the complete poem follows. Verify which lines you have already added. As you append the lines, do so carefully, as before, adding only a few lines at a time. Each time you change the file, save it, right click on the file name, and select Run File to see the results in the applet viewer. At this point, do not view the output in the HTML file.

Robinson Jeffers Inside a cave in a narrow canyon near Tassajara The vault of rock is painted with hands, A multitude of hands in the twilight, a cloud of men's palms, no more, No other picture. There's no one to say Whether the brown shy quiet people who are dead intended Religion or magic, or made their tracings In the idleness of art; but over the division of years these careful Signs-manual are now like a sealed message Saying: \"Look: we also were human; we had hands, not paws. All hail You people with the cleverer hands, our supplanters In the beautiful country; enjoy her a season, her beauty, and come down And be supplanted; for you also are human.\"

9. Modify your source code to allow for as many lines as there are in the poem:

outputArea = new JTextArea(1, 60);

10. **Modify** your HTML file so that it can accommodate the size of the generated applet. To accomplish this, access the HTML file, as before. Locate the width and height properties. Change them to 700 and 400, respectively, as follows:

<APPLET codebase="classes" code="FirstJApp.class" width=700 height=400></APPLET>

Viewing the HTML

- 1. Create a folder on your desktop named **TestHTML**.
- 2. Open Windows Explorer and navigate o the build directory of the 0010JApp project folder.
- 3. Copy the classes directory and the DirsJApp.html file to the **TestHTML** folder..
- 4. Paste the folder and the *****.**HTML** file into the **TestHTML** folder.
- 5. Double click on the *****.**HTML** file in the **TestHTML** folder and see it execute in the browser.

- 6. If you do not see all the poem, right click on the *****.**HTML** file in the **TestHTML** folder and select Open with WordPad. (WordPad allows you to open your files with line returns. It is located in WindowsNT/Accessories.)
- 7. Modify the APPLET height property so that the applet is 450 pixels high:

```
<APPLET codebase="classes" code="FirstJApplet.class" width=450
height=450></APPLET>
```

8. Test again. Now the applet opens and shows the whole image. If all the lines are not showing, modify the height attribute once again.

Opening and Closing Projects

1. Working in the Projects panel, right click successively on the two projects you have created thus far and select Close.

Projects	€ × Files	
= 🍉 卪	010JApp	
÷	New	•
⊡ …[Build	
😐 🗲	Clean and Build	
Ė€	Clean	
	Generate Javadoc	
	Run	
	Debug	
	Profile	
	Test	Alt+F6
	Set Configuration	•
-	Set as Main Project	
	Open Required Projects	
	Close	

2. Having closed the two projects, reopen them. NetBeans does not have a given file type for its projects. Instead, it identifies projects according to the folders that contain the files. Accordingly, when you want to open a NetBeans file, select Project > Open, and navigate to the folder that contains you NetBeans project folders. Click on the folder to open the project. See the following figure.

🗊 Open Proje	t in the second s	$\mathbf{\times}$
My Recent Documents	Look in: Aava0100 Java0100 Java0100 ConsoleWork JavaJApp ConsoleWork ConsoleW	
My Documents		
My Network Places	File name: Intro Fall 2010\Java Lab Project Files\0100JavaLab\Project Files\Java0100\0010JApp Open Project Files of type: Project Folder Cancel	

- 3. The project opens and you can then begin working on it.
- 4. Prior to terminating your user session, close all your projects and transfer the project folders to your portable storage device.

Other Topics to Review

- 1. If you are going to rename a file in NetBeans, right click on the name of the file and select Refactor.
- 2. To format code, select Source > Format.
- 3. To view compiler errors, select Window > Tasks. The Tasks panel shows you specific problems with your code. Click on a line in the tasks list, and you will be taken to the problem line.
- 4. To see the toolbar that allows you to click to insert comments, select View > Show Editor Toolbar.
- 5. If you want to see the difference between two versions of a file, open one of the files, and select Tools > Diff.
- 6. To copy your project, close the project in NetBeans and then copy the project folder to your flash drive. Even if you close out of NetBeans without closing a project before moving it, no harm will be done. When NetBeans opens, it resets its properties so that any projects that it no longer tried to open moved or deleted projects.