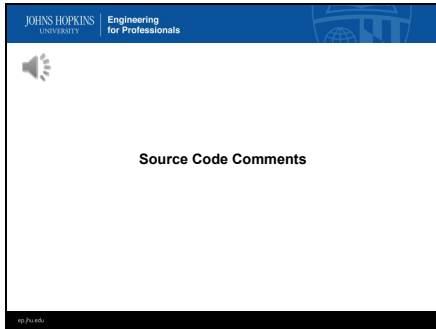
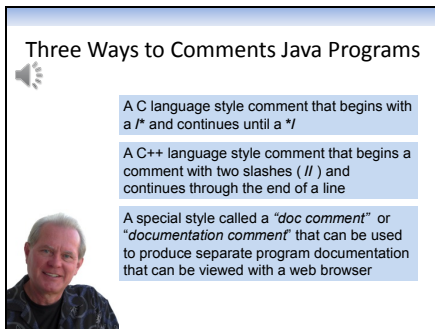


1



In this lecture you will learn how to add comments to your source code.

2



It is a very good practice to add comments to your Java programs.

Comments add explanatory information that helps to make your programs easier to read and understand. Comments are important for people who read your programs. The Java compiler doesn't care about comments and will not compile comments into byte-code.

Source code comments are sometimes called inline program documentation by professional programmers.

There are 3 ways to add comments to Java programs...and we will discuss and use two of them in this course.


The first way to add comments is to use what is commonly called a C-style comment, because it is how comments are added in the C programming language. Java borrowed this style from that language. A C-style comment begins with a `/*` and ends with a `*/`. Any information that is contained between the `/*` and the `*/` is considered to be a comment by the Java compiler.

The second way to add comments is to use a C++-style comment, that Java borrowed from the C++ programming language. Using this style, a comment begins with a `//` and continues through the end of a line. This style is used primarily for comments that span a

single line or part of a line.

The third way of adding comments to a program is to use something called a “doc comment”. This type of comment is used to produce external documentation for a program that can be printed out or published on a web page. We will not discuss “doc” comments in this course.

3



Using C-Style Comments

```
/* This is a C-style comment. */  
/*****  
 * This style of comment is particularly useful *  
 * When your comments are lengthy and span many *  
 * lines. *  
 *****/  
  
public class MyProgram  
{  
    public static void main( String [] args )  
    {  
        int interestRate;    /* Periodic interest rate */  
        -  
        /* Calculate the year-end account balance */  
        -  
    }  
}
```


Let's take a look at using C-style comments first.

C-style comments treat all text between a `/*` and a `*/` as a comment. This style is particularly useful when you have lengthy comments.

Comments can be placed virtually anywhere in your program. The most common place to add comments is at the top of a source file, to the right of a variable declaration, and at the beginning of a group of program statements that perform some type of computation.

In this example, I've shown the comments using a different color only because I wanted them to stand out so that you can more easily recognize them.

4



Using C++-Style Comments

```
// This is a C++-style comment.  
// This style of comment begins with two slashes  
// and is in effect until the end of the line  
  
public class MyProgram  
{  
    public static void main( String [] args )  
    {  
        int interestRate;    // Periodic interest rate  
        -  
        // Calculate the year-end account balance  
        -  
    }  
}
```

The C++-style comment is designed to comment a single line or portion of a line.

A double-slash marks the beginning of the comment, and everything to the right of the slashes, to the end of the line, is a comment.

In this example, I've shown the comments using a different color again only because I wanted them to stand out so that you can more easily recognize them.

5



Using Both Styles of Comments

```

/*****
 * This style of comment is particularly useful *
 * When your comments are lengthy and span many *
 * lines. *****/

public class MyProgram
{
    public static void main( String [] args )
    {
        int interestRate;    // Periodic interest rate

        // Calculate the year-end account balance
        ...
    }
}

```

It is perfectly okay to use both styles of comments in your programs, as you see illustrated here.

Many programmers use the C-style comment when comments span multiple lines and use the C++-style for shorter comments. The choice is up to you.

6



Creating Documentation With javadoc

Class AddDemo	
java.lang.Runtime	
public class AddDemo	
Package: SumDemo	
This class computes and displays the sum of two integers using a class method called sum().	
Version:	1.0
Author:	The Java Master
Constructor Summary	
Constructor	
Method Summary	
Method:	sum(int firstValue, int secondValue)
The sum() method defines the integers to be summed, passes them to the sum() method, and displays the result returned by sum().	
Method:	main(String [] args)
This method creates the sum of two integers.	

A third way to document your Java code is to use a tool called javadoc that comes with the JDK.

Sun provides extensive online documentation for every Java class in the form of HTML documents that have a relatively standard format and can be read by any web browser.

The javadoc tool is used to create similar documentation for your own Java code.

A partial example of such documentation is shown here. The complete web page can be found in the course downloads folder. Just load the page into any web browser to view it.

7



Documentation Comments

```

/** starts a documentation comment
/**
 * This class computes and displays the sum of two integers
 * using a class method called sum().
 * @author The Java Master      tag
 * @version 1.0
 * @param firstValue the first value to be summed
 * @param secondValue the second value to be summed
 */
 */ ends a documentation comment

@author gives the name of the author
@version gives the version number of the code
@param gives the name & description of a parameter used by the code
@return gives a description of the value returned by a method

```

To use the javadoc tool a programmer must add special comments, called documentation comments, into the source code.

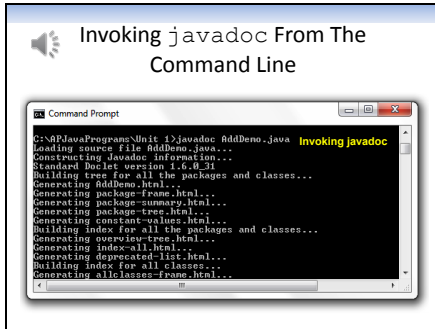
A `/**` starts a documentation comment and a `*/` ends a documentation comment as illustrated here.

Documentation comments may also include tags .

Each tag must appear on a separate line and includes special information...such as an author name or version number of the code, as illustrated here.

Four commonly used tags are `author`, `version`, `param`, and `return`.

8



The javadoc utility can be invoked from the command line...or from within many Interactive Development Environments.

The simplest way to invoke javadoc is to follow the command with the file names to be documented. There are also numerous options that can be specified.

This example shows how to invoke javadoc from the command line. Note that there are numerous HTML files generated by this program. They will be written to a directory called doc by default.

A complete example of using javadoc can be found on the course website.