CS330: Programming Language Project (PLP)
Assignment 2: Installation, programming environment, and Hello, World!

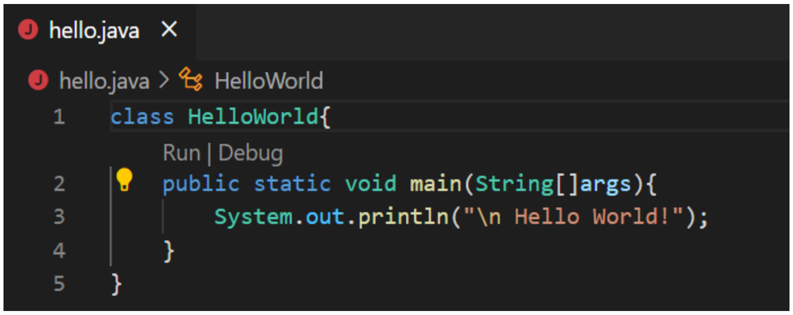
Now that you've picked a language and learned about its history and uses, it's time to actually get it set up and use it. For this assignment you must:

Install your programming language and anything else that it needs to run (a programming environment or something similar)

1. Write a "hello world" program in that language (checking the Internet for one is fine)
2. Run the program
3. Be prepared to show me that you can run the program during lab

hello.java

hello.java located in my **GitHub repository**[1]



```
class HelloWorld{
    public static void main(String[]args){
        System.out.println("\n Hello World!");
    }
}
```

Code Analysis:
class – creates a class, in this case called “HelloWorld.”

public static void main() – the main start up method of a java program. It makes it visible to all, static, is not returning any value
(i.e. public, static void).

String[]args – command line argument

System.out.println() – prints a statement to the terminal

Your write-up should address the following questions. Try to answer them in such a way that someone else would be able to follow your instructions and run your program (because someone probably will...):

1. Can this language be installed on any operating system (Windows, Mac, Unix/Linux)? If not, what are its limitations?

    Java can be installed on any operating system such as **Microsoft Windows, Solaris, Linux, and macOS**. [2]

[1] Lopez E, *hello.java*, (2020), GitHub repository,
https://github.com/elianalopez/Java/blob/master/Assignment2/hello.java

[2] Oracle. (n.d.). Oracle JDK 8 and JRE 8 Certified System Configurations. Retrieved September 09, 2020, from
https://www.oracle.com/java/technologies/javase/products-doc-jdk8-jre8-certconfig.html

2. Give instructions for how to install the language

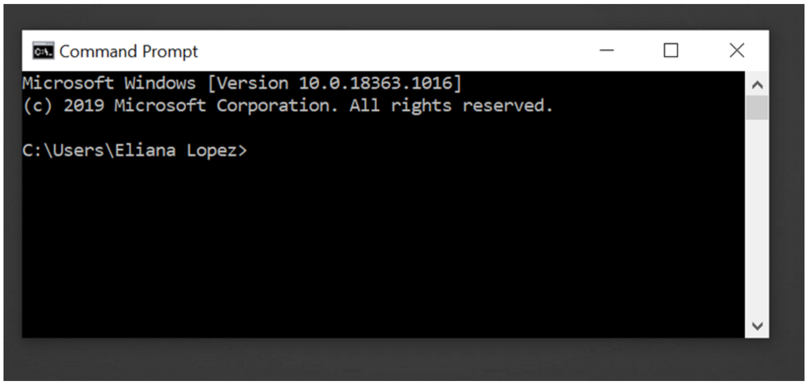
See my GitHub for a full detailed response[3]

How to Install Java for Windows:

For a brief overview of my **detailed response**, am going to separate this lesson unto 4 parts: Before Installing, Installing, Checking, and Creating Environmental Variables.

Before Installing:

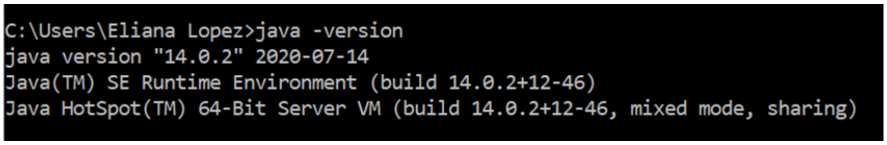
Before installing Java, you should check to see if your computer has Java to begin with. How to check is by going to the *Command Prompt* application, by searching for it in the Start bar.



```
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Eliana Lopez>
```

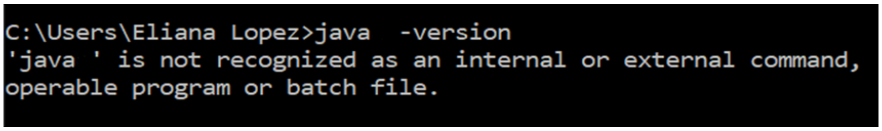
This what the Command Prompt application looks like

Once the Command Prompt application is open, input, ‘`java -version`’ and you are bound to get either one of these responses as a result:



```
C:\Users\Eliana Lopez>java -version
java version "14.0.2" 2020-07-14
Java(TM) SE Runtime Environment (build 14.0.2+12-46)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.2+12-46, mixed mode, sharing)
```

This will be the output if your computer has Java installed



```
C:\Users\Eliana Lopez>java  -version
'java ' is not recognized as an internal or external command,
operable program or batch file.
```

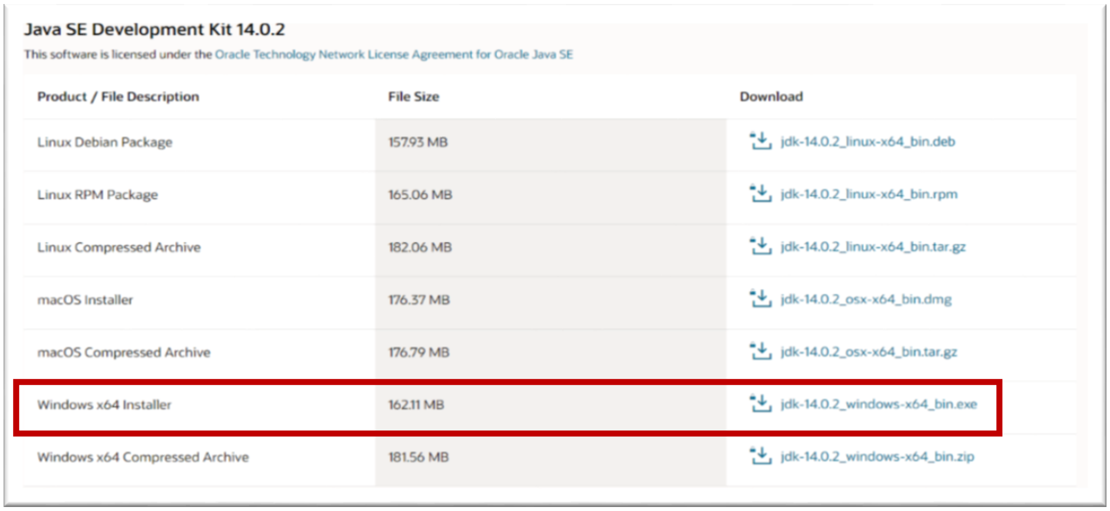
This will be the output if your computer does not have Java installed.

If your output is similar to the second image, we can continue on to the installation process of this tutorial, if not you are all set!

[3] Lopez E, Java_Installation, (2020), GitHub repository, https://github.com/elianalopez/Java/blob/master/Assignment2/Java_Installation.pdf

Installing:

To download Java you must go to Oracle's Java SE Development Kit download page and Scroll down to download the Windows x64 Installer. It should be an executable file, **.exe**. I have attached an image and a description below of what this step should look like.



Java SE Development Kit 14.0.2

This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE

Product / File Description	File Size	Download
Linux Debian Package	157.93 MB	jdk-14.0.2_linux-x64_bin.deb
Linux RPM Package	165.06 MB	jdk-14.0.2_linux-x64_bin.rpm
Linux Compressed Archive	182.06 MB	jdk-14.0.2_linux-x64_bin.tar.gz
macOS Installer	176.37 MB	jdk-14.0.2_osx-x64_bin.dmg
macOS Compressed Archive	176.79 MB	jdk-14.0.2_osx-x64_bin.tar.gz
Windows x64 Installer	162.11 MB	jdk-14.0.2_windows-x64_bin.exe
Windows x64 Compressed Archive	181.56 MB	jdk-14.0.2_windows-x64_bin.zip

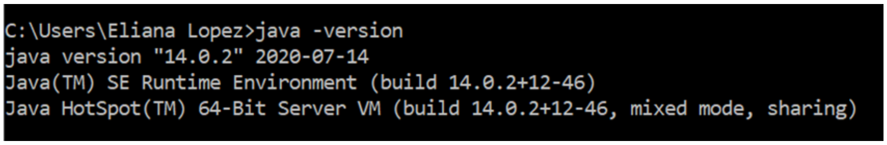
This is an image of what the page should look like and what is boxed is the file you should download

Checking:

There are two ways to check if you have successfully installed Java, through the Command Prompt or Via your program files.

Command Prompt:

Checking via the command prompt is similar to what we have done during the "Before Installing" part of this tutorial. You should open the Command Prompt once again, type '*java -version*' and your response should look like this!

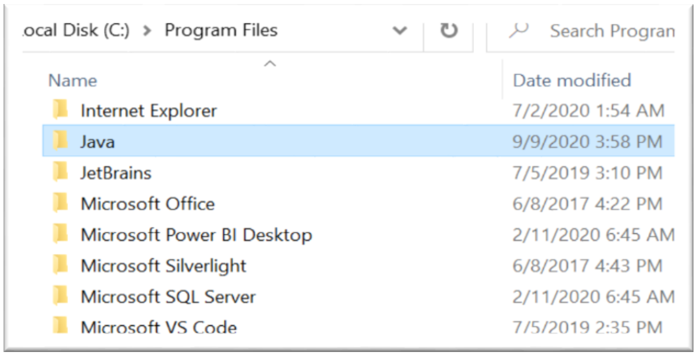


```
C:\Users\Eliana Lopez>java -version
java version "14.0.2" 2020-07-14
Java(TM) SE Runtime Environment (build 14.0.2+12-46)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.2+12-46, mixed mode, sharing)
```

Java is successfully installed!

Program files:

The other way to check is via your program files which is located in your C drive **(C:)**, or any drive you downloaded Java in. When you check your program files you should find a folder named *Java* and in that folder another folder named *jdk*, which has all the respect folders and files that help Java operate properly!



Name	Date modified
Internet Explorer	7/2/2020 1:54 AM
Java	9/9/2020 3:58 PM
JetBrains	7/5/2019 3:10 PM
Microsoft Office	6/8/2017 4:22 PM
Microsoft Power BI Desktop	2/11/2020 6:45 AM
Microsoft Silverlight	6/8/2017 4:43 PM
Microsoft SQL Server	2/11/2020 6:45 AM
Microsoft VS Code	7/5/2019 2:35 PM

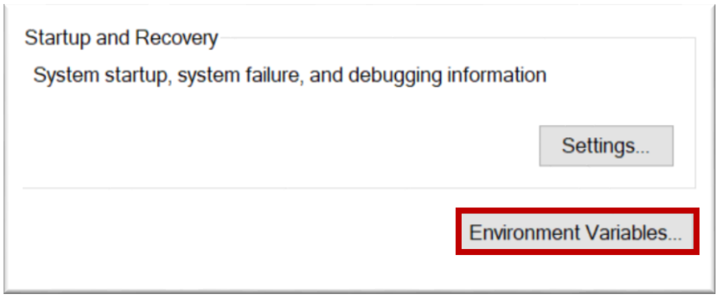
This the is Java folder located in your program files directory

Creating Environmental Variables:

Now that we know that Java is installed we have to set our environmental variables for Java, which would make things easier to run! What setting the environmental variables would do is provide a path for compiling a Java Program.

To start we must search "*Environmental Variables*" in the Start menu and click in the panel that says "*Edit the system environment variables*."

A System Properties window will open and then you must find the Startup and Recovery Section and then click the *Environment Variables...* button


Startup and Recovery

System startup, system failure, and debugging information

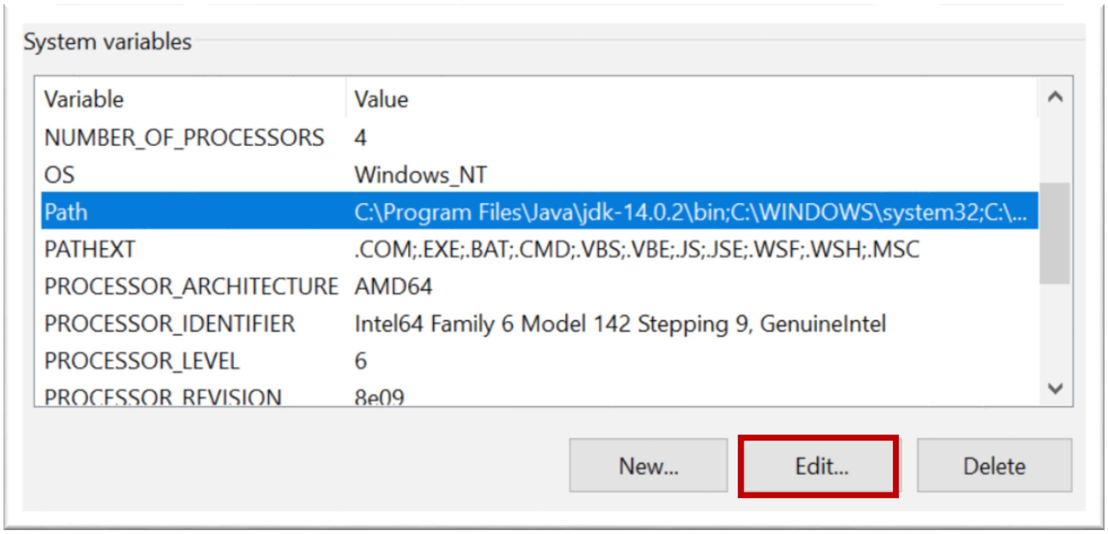
Click *Environment Variables...*

You would then search for *Path* in the Systems Variable section of the window (see **Image A**) *click Edit...,* click *new***,** and copy paste the file directory that the Java folder is located, go to the JDK folder and you should have something like this.

C:\Program Files\Java\jdk-14.0.2\bin

You then copy and paste this to the empty row and you should have something like **Image B**. Then click okay and you should be back in the *Environmental Variables* window once again. Once you are in the window again you will click new this time, just like in **Image C**.

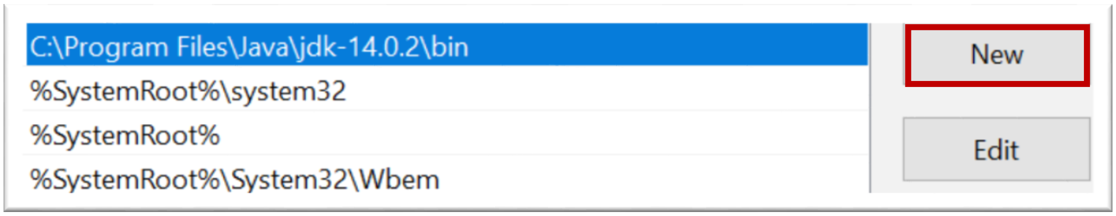
Image A



Variable	Value
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Java\jdk-14.0.2\bin;C:\WINDOWS\system32;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 9, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8e09

Go to Path and click on the *Edit...* Button

Image B



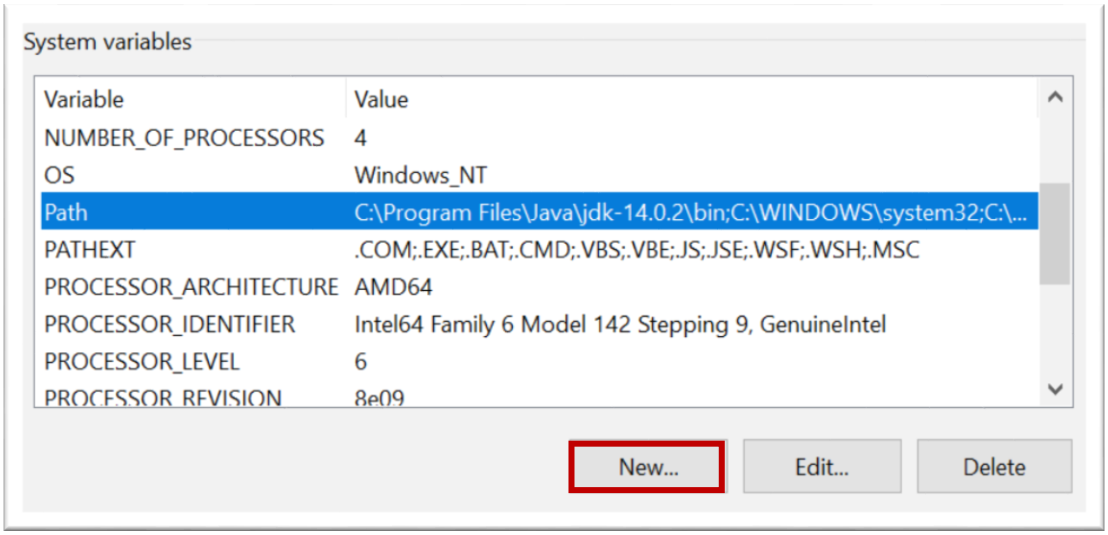
C:\Program Files\Java\jdk-14.0.2\bin
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem

Click the New Button and paste "*C:\Program Files\Java\jdk-14.0.2\bin*" from the file window

After you click new you are directed to a new window that pops called *Edit System Variable* out and you fill the two text columns**,** *Variable name* and *Variable Value* as seen in **Image D**. Variable name would be called "JAVA_HOME" and variable value would be a copy paste of the Java file directory all the way up to the JDK folder, *C:\Program Files\Java\jdk-14.0.2.*

When these columns are filled out you click on *OK* and you are ready to start programming with Java! After all of these steps you should have successfully have the program installed in your computer!

Image C

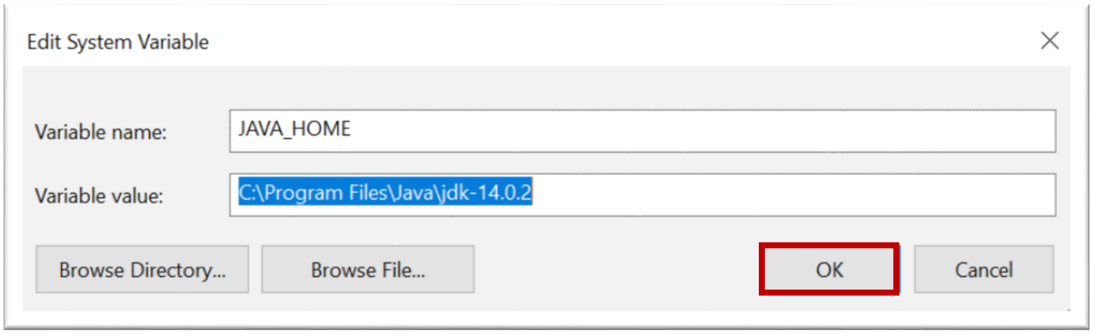


System variables

Variable	Value
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Java\jdk-14.0.2\bin;C:\WINDOWS\system32;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 9, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8e09

Click *New...*

Image D



Edit System Variable

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk-14.0.2

Then name the variable JAVA_HOME and copy paste the variable value from the file window

Variable Value: *"C:\Program Files\Java\jdk-14.0.2"*

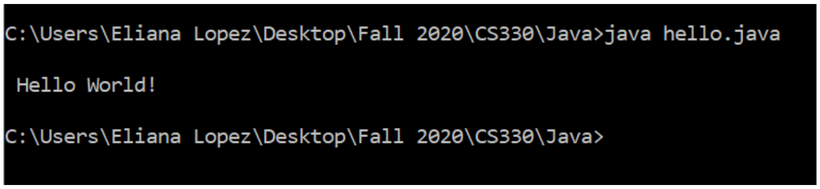
Then click *OK* and you are all done!

3. Where do you write programs in this language (as in: in a text editor, a special editor just for that language, something else?)
    a. Related: Does this language come with a recommended programming environment? What is it? If not, how did you pick the one that you'll be using?

    I like to write my languages in a text editor called **Visual Studio Code** because it is an elegant editor with the ability to make it "my own" with its use of extensions. Visual Studio Code can also be used with a variety of programming languages such as Java, so I naturally gravitated my use of writing Java into this editor. However other popular editors for Java include Eclipse, IntelliJ IDEA, and NetBeans[4].

    Once downloaded, you will notice that **Java does not come with a programming environment**, the simplest way to program in Java is by using notepad, saving the file under .java and running it from the command line by invoking "*`java scourcecode.java.`*"

    I have inserted an example of a Java file running from the command line down below.

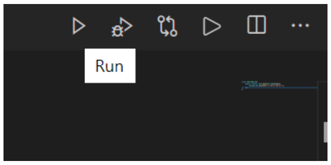


```
C:\Users\Eliana Lopez\Desktop\Fall 2020\CS330\Java>java hello.java

 Hello World!

C:\Users\Eliana Lopez\Desktop\Fall 2020\CS330\Java>
```

4. How do you run programs that you write?

    I run a program typically through Visual Studio Code or some type of IDE. If not I can also run it through the command prompt by invoking it, such as with the example above.

    How I run a program through Visual Studio Code is by clicking on the run icon from the top right. I have inserted an image of what a run button typically looks like



5. Is there a lot of boiler-plate code that you need to write a program (like in Java)? Or can you just start writing (like in Python and Perl)?

    There is plenty of boiler-plate code in Java. Boiler-plate code is code that requires phrases of code to be included, with or without slight adjustments.[5] Boiler-plate code helps with compiling the program.
    An example of boiler-plate code:

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

    This line is a way to invoke a command line argument as an output.

[4] Singh, V. (2020, August 20). Best Java IDE 2020: Most Popular Java IDE for Coding. Retrieved September 10, 2020, from https://hackr.io/blog/best-java-ides

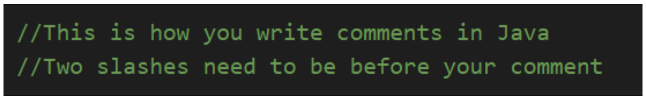
[5] What is a boilerplate code? (n.d.). Retrieved September 10, 2020, from https://www.educative.io/edpresso/what-is-a-boilerplate-code

6. How do you write comments in your language?

In Java there are three ways to write comments, single line, multi-line, and documentation comments.[6]

Single line:

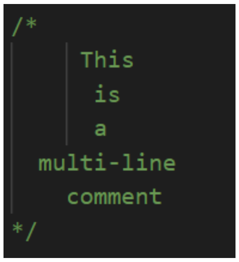
Single line comments are one lined comments that are executed one two slashes (//) are written. Anything prior to the two slashes are not included within the comments but anything typed after is included. There is an illustration that demonstrates single lined comments in Java.



```
//This is how you write comments in Java
//Two slashes need to be before your comment
```

Multi-line:

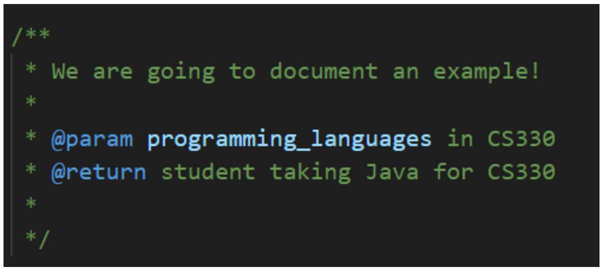
Multi-line comments are comments for multiple lines of code. How It starts is a slash and a star (/*) and ends with a star and a slash (*/). Anything between the start and the end is included within the comments. There is an illustration that demonstrates multi-line lined comments in Java.



```
/*
    This
     is
     a
  multi-line
    comment
*/
```

Documentation Comments:

Documentation comments are similar to multi-line comments however these comments go in depth regarding the functionality of your program. A documentation comment begins with a slash and two stars (/**) and ends with a star and slash (*/). Each comment between the start and end begin with a start as well! There is an illustration that demonstrates documentation comments in Java.



```
/**
 * We are going to document an example!
 *
 * @param programming_languages in CS330
 * @return student taking Java for CS330
 *
 */
```

[6] Comments in Java. (2017, April 24). Retrieved September 10, 2020, from https://www.geeksforgeeks.org/comments-in-java/

The answers to these questions will be put in a GitHub repository that you create, which you will send me a link to. Make sure that your answers are clear, accurate, and fully-formed: remember that these tutorials are public, and GitHub users don't have the context of the assignment that you do.

Explain the reasoning behind the answers as much as possible. If there is no clear-cut answer to a question, explain why not. And cite your sources!

A sample GitHub tutorial (for Perl) can be found here: https://github.com/amber-stubbs/PerlTutorial