

COMP249 Tutorial 1: Command Line Basics: Compile and Launch Java Programs Using the Command Line

Ella Noyes

July 11, 2024

1 How to Access the Command Line Interface

The command line is just another way of interacting with programs. Nowadays, a lot can be done using a Graphical User Interface (GUI), but the Command Line Interface (CLI) has its own advantages.

I'll walk you through this live during the tutorial, but here is a transcript to use as a reference if you have any issues.

- Windows Users: Search “Command Prompt”.
- MacOS Users: Search “Terminal”.
- Linux Users: Name depends on which distribution you’re using. Search “Terminal”, or “Command”, or “Prompt” and see what comes up. Or try the Ctl + Alt + T shortcut.

2 Getting Started With the Command Line

Open the command line interface. Checking Assignment 0 using DebugRunner requires Java 19 or above. Start by checking which version of Java you’re working with.

In the command line, enter:

- Windows Users:

```
java —version
```

- MacOs and Linux Users:

```
java –version
```

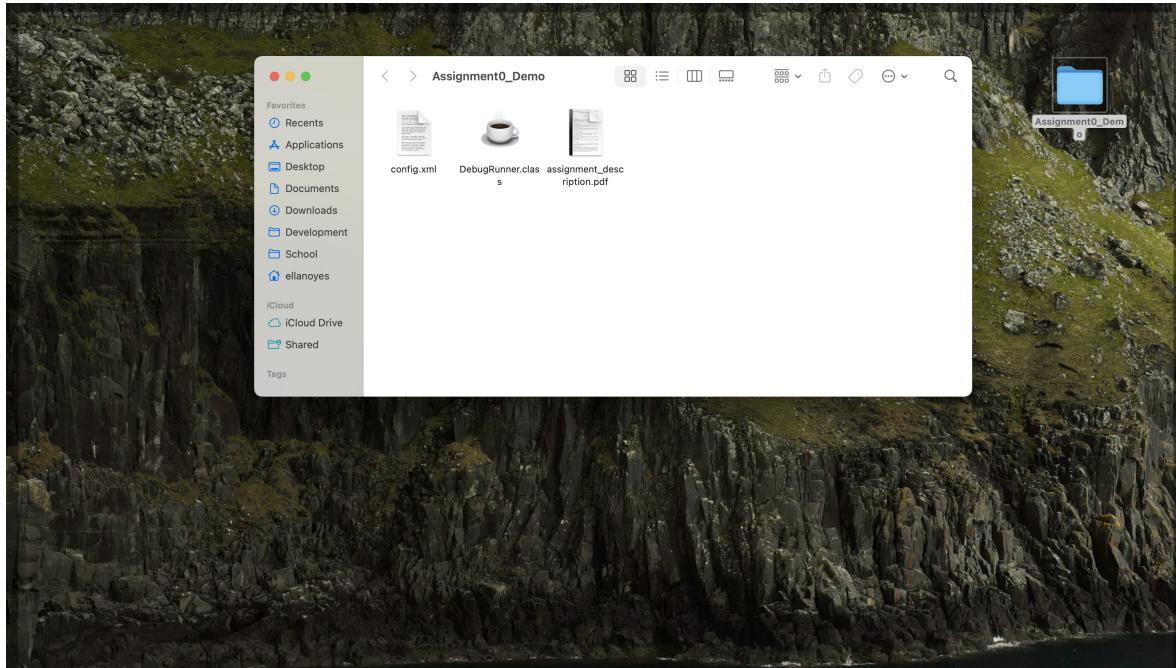
```
[ellanoyes@Ellas-MBP-3 ~ % java -version
openjdk version "19.0.2" 2023-01-17
OpenJDK Runtime Environment Temurin-19.0.2+7 (build 19.0.2+7)
OpenJDK 64-Bit Server VM Temurin-19.0.2+7 (build 19.0.2+7, mixed mode, sharing)
ellanoyes@Ellas-MBP-3 ~ %
```

As you can see, I have Java 19.0.2 installed and set as the default Java runtime.
Troubleshooting:

- If you have several versions of Java installed, then this command will state the default Java runtime being used. You may need to change this. If so, search “Change JAVA_HOME variable” online, specifying the operating system you’re using, and follow instructions.
- If you do not have Java 19 or above installed, then search “Java 19” online and download an installer from a trusted distributor (e.g., Oracle). Follow the steps provided by the installer.

3 Opening the Assignment 0 Directory In the Command Line

In this demonstration, I have downloaded Assignment 0 from Moodle and dragged it to my Desktop. A directory is also known as a folder: it contains references to files and sometimes to other directories. From now on, I will refer to a folder as a directory.



Your assignment directory contains three files: DebugRunner.class (a compiled Java class called DebugRunner), config.xml (used by DebugRunner to test your assignment), and an assignment description pdf. We're opening them using a GUI here. So let's see what that looks like in the CLI.

I have the Assignment 0 directory in my Desktop directory. When I open the command line, I need to navigate to the Desktop directory, then to my Assignment0_Demo directory. Introducing your first command line commands: Change Directory “cd” (“dir” for Windows Users!), and List “ls”

```
cd Desktop  
ls
```

ls will list the contents of your current directory

```
Last login: Wed Jul 10 08:29:50 on ttys002  
[ellanoyes@Ellas-MBP-3 ~ % cd Desktop  
[ellanoyes@Ellas-MBP-3 Desktop % ls  
Assignment0_Demo           assignment_directory.png  
ellanoyes@Ellas-MBP-3 Desktop %
```

My Desktop directory contains a screenshot from earlier and a directory “Assignment0_Demo”. I’ll open that up

```
cd Assignment0_Demo  
ls
```

```
Last login: Wed Jul 10 08:29:50 on ttys002  
[ellanoyes@Ellas-MBP-3 ~ % cd Desktop  
[ellanoyes@Ellas-MBP-3 Desktop % ls  
Assignment0_Demo assignment_directory.png  
[ellanoyes@Ellas-MBP-3 Desktop % cd Assignment0_Demo  
[ellanoyes@Ellas-MBP-3 Assignment0_Demo % ls  
DebugRunner.class assignment_description.pdf config.xml  
ellanoyes@Ellas-MBP-3 Assignment0_Demo %
```

Just like Finder or File Explorer, the CLI displays the files and directories within a directory when you enter ls.

3.1 Following the Assignment Instructions

HelloWorld Directory Structure

```
myassignment/bin/  
  DebugRunner.class  
  config.xml  
a0/  
  HelloWorld.class
```

The assignment description gives us a directory structure to follow. Our HelloWorld.java class should be in a package called a0. So we’ll need to create a directory called a0, then a Java class within that directory called HelloWorld.java

In my open assignment directory, I’ll make a directory called a0 using the “mkdir” (make directory) command. So still in the Assignment0_Demo directory, enter

```
mkdir a0  
cd a0
```

```
[ellanoyes@Ellas-MBP-3 Assignment0_Demo % mkdir a0  
[ellanoyes@Ellas-MBP-3 Assignment0_Demo % ls  
DebugRunner.class a0 assignment_description.pdf config.xml  
[ellanoyes@Ellas-MBP-3 Assignment0_Demo % cd a0  
[ellanoyes@Ellas-MBP-3 a0 % ls  
ellanoyes@Ellas-MBP-3 a0 %
```

I have created an empty directory a0, then used cd to change into that directory. Time to write our Java program!

3.2 Creating HelloWorld.java

Now that we're in a0, let's create a text file called HelloWorld.java.

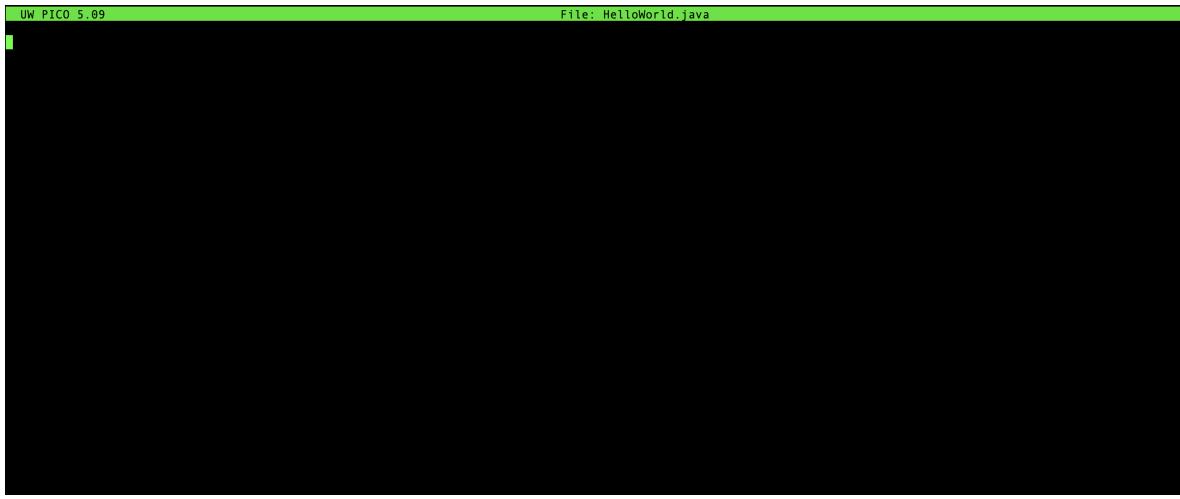
- Windows Users:

```
notepad HelloWorld.java
```

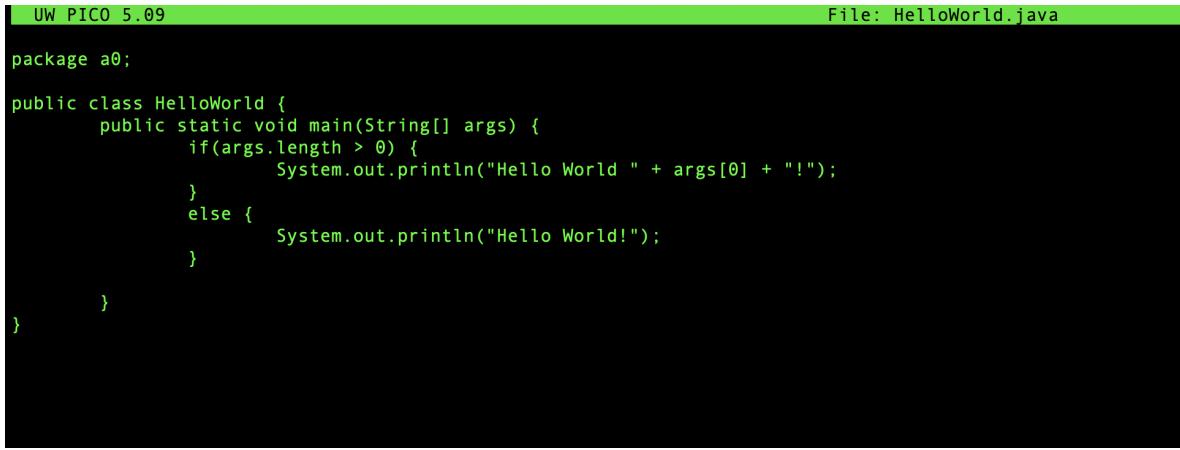
- MacOS and Linux Users:

```
nano HelloWorld.java
```

nano (for MacOS and Linux users) is a text editor. Windows users can use notepad.

A screenshot of a terminal window titled "UW PICO 5.09". The title bar also displays "File: HelloWorld.java". The main area of the terminal is a black text editor window with a single character "I" at the top left, indicating it is ready for input. There is no code or text visible in the editor area.

nano just looks like this, it's not much to look at. Now it's business as usual, you can write out your HelloWorld program. Do this yourself! Here's my code, but you'll learn something if you write it out yourself. Using an IDE is much smoother than a text editor as it checks syntax and makes suggestions. But knowing how to do this in a text editor is a useful skill!

A screenshot of a terminal window titled "UW PICO 5.09". The title bar also displays "File: HelloWorld.java". The main area of the terminal shows the following Java code:

```
package a0;

public class HelloWorld {
    public static void main(String[] args) {
        if(args.length > 0) {
            System.out.println("Hello World " + args[0] + "!");
        } else {
            System.out.println("Hello World!");
        }
    }
}
```

The code is displayed in green and black text on a black background.

Compile your code and fix any compilation errors by returning to the text editor. To compile a java program, enter “javac” (java compile) followed by your Java file name “HelloWorld.java”. I didn't get any compilation errors (these will display in the command line, so things are looking alright for now!

```
javac HelloWorld.java
```

```
[ellanoyes@Ellas-MBP-3 a0 % nano HelloWorld.java
[ellanoyes@Ellas-MBP-3 a0 % javac HelloWorld.java
[ellanoyes@Ellas-MBP-3 a0 % ls
HelloWorld.class      HelloWorld.java
ellanoyes@Ellas-MBP-3 a0 %
```

If you now list the contents of your a0 directory, you'll see your HelloWorld.java, and a new file called HelloWorld.class. The Java compiler has processed HelloWorld.java and produced a new file containing Java bytecode. This file will be used by the Java Virtual Machine when you go on to launch your program!

```
javac HelloWorld.java
```

Let's try launching HelloWorld to see what comes out. (Start by commenting out "package a0;" and recompiling, you'll need this later but for now we're just testing our program! To launch your Java program, enter "java HelloWorld [arguments]". Replace [arguments] with any String arguments that you want to pass to HelloWorld's main() method.

```
java HelloWorld.java
```

```
[ellanoyes@Ellas-MBP-3 a0 % nano HelloWorld.java
[ellanoyes@Ellas-MBP-3 a0 % javac HelloWorld.java
[ellanoyes@Ellas-MBP-3 a0 % java HelloWorld
Hello World!
[ellanoyes@Ellas-MBP-3 a0 % java HelloWorld Ella
Hello World Ella!
ellanoyes@Ellas-MBP-3 a0 %
```

4 Testing your Assignment Using DebugRunner

Now, un-comment "package a0;" and recompile your program—DebugRunner will expect to find HelloWorld in this "a0" package.

Now return to the "Assignment0_Demo" directory

```
cd ../
```

Using the cd command with ../ changes to the current directory's parent directory (i.e., "Assignment0_Demo"). Similarly, entering "cd ../../" would navigate to the current directory's parent directory's parent directory.

```
[ellanoyes@Ellas-MBP-3 a0 % nano HelloWorld.java
[ellanoyes@Ellas-MBP-3 a0 % javac HelloWorld.java
[ellanoyes@Ellas-MBP-3 a0 % cd ../
[ellanoyes@Ellas-MBP-3 Assignment0_Demo % ls
DebugRunner.class          a0                      assignment_description.pdf      config.xml
ellanoyes@Ellas-MBP-3 Assignment0_Demo %
```

Notice that DebugRunner has the .class extension—DebugRunner.java been compiled. Launch DebugRunner to test and grade your assignment.

```
java DebugRunner
```

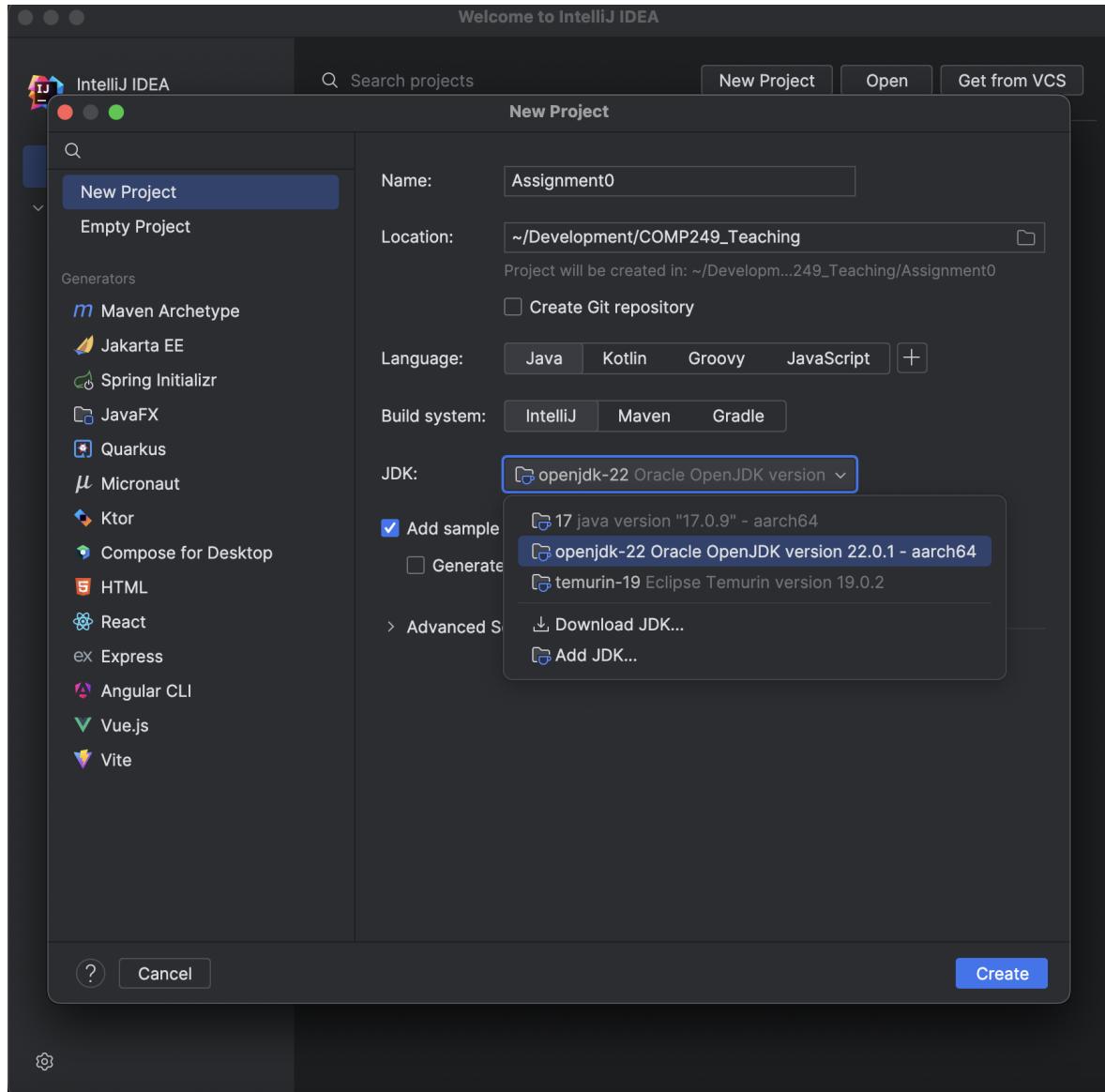
```
[ellanoyes@Ellas-MBP-3 Assignment0_Demo % java DebugRunner  
file:/Users/ellanoyes/Desktop/Assignment0_Demo/DebugRunner.class  
Assignment 0: Hello World!  
Testing for grade weight: 80.0  
    can you sysout?  
        Expecting: "Hello World 40064416" and found it. (+80.0)  
Testing for grade weight: 10.0  
    can you read carefully?  
        Expecting: "Hello World 40064416!" and found it. (+10.0)  
  
Assignment 0: Hello World CommandLine!  
Testing for grade weight: 10.0  
    can you use commandline?  
        Expecting: "Hello World Person!" and found it. (+10.0)  
  
100.0  
ellanoyes@Ellas-MBP-3 Assignment0_Demo %
```

If you don't get 100.0/100.0, return to HelloWorld.java and try to find the issue! Always check your score with DebugRunner before submitting an assignment for this course!

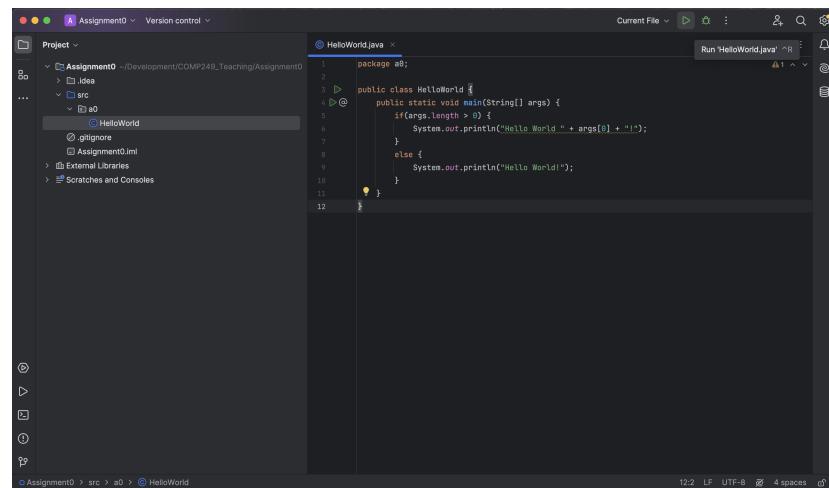
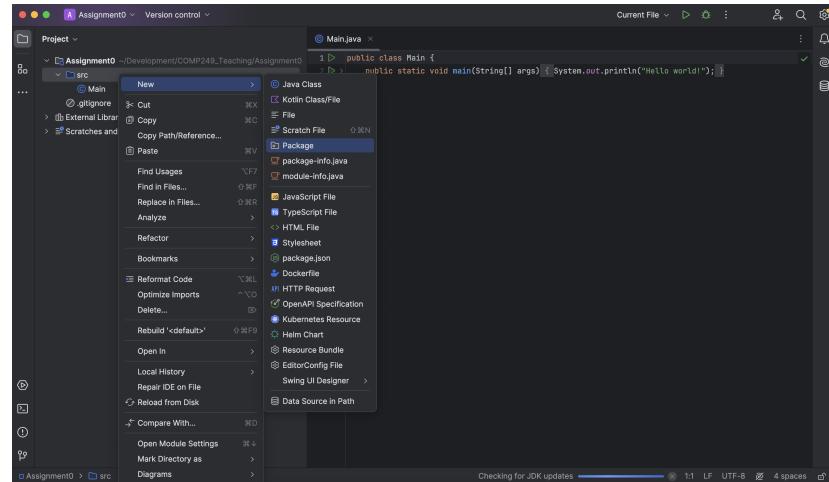
5 Hello World in IntelliJ

To do all of this in IntelliJ:

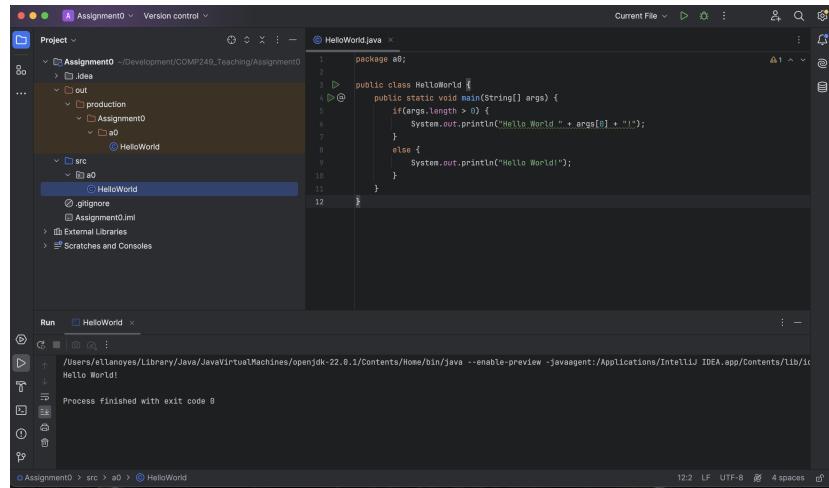
1. Create a new project in IntelliJ. For the JDK, choose Java 19 or above. If you don't have Java 19 or above installed, then select Download JDK... and download a newer version of Java.



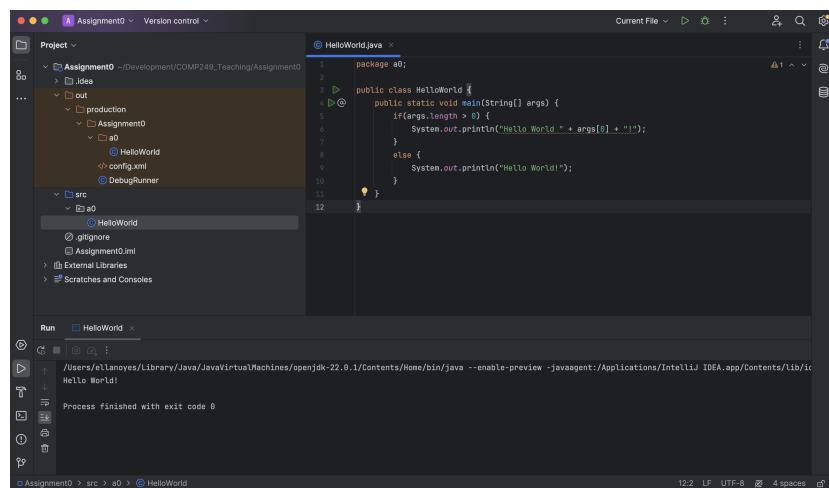
2. In this new project, create a package “a0” by right clicking “src” and selecting New > Package. Drag the example Main class into this new package and confirm the refactoring. Rename this class “HelloWorld” and write your code for this assignment in the class’ main() method.



3. Press the green arrow to run your program. This will compile and launch your Java program. The Java bytecode file that the compiler produces will be placed in a new directory called “out”.



4. DebugRunner.class and config.xml should be placed in the same directory as the a0 package. Drag those files into your project



5. Click on the terminal bottom on the bottom left corner of your screen

The screenshot shows the IntelliJ IDEA interface. The project structure on the left includes a 'Production' folder containing 'Assignment0', 'out', 'a0', and 'config.xml'. The 'src/a0' package contains a 'HelloWorld' class. The 'Terminal' window at the bottom shows the command `elarnoyes@Ellas-MBP-3 Assignment0 %`.

6. In terminal, open the directory containing a0, DebugRunner.class, and config.xml using the cd command. Run DebugRunner by entering java DebugRunner (as we did in the command line approach above).

The screenshot shows the IntelliJ IDEA interface with the same project structure. The terminal window now displays the output of the Java application:

```
elarnoyes@Ellas-MBP-3 Assignment0 % cd out/production/Assignment0
elarnoyes@Ellas-MBP-3 Assignment0 % java DebugRunner
file:/Users/elarnoyes/Development/COMP249_Teaching/Assignment0/out/production/Assignment0/DebugRunner.class
Assignment 0: Hello World!
Testing for grade weight: 80.0
    can you sysout?
        Expecting: "Hello World 400b4416" and found it. (+80.0)
Testing for grade weight: 10.0
    can you read commandLine?
        Expecting: "Hello World 400b441d" and found it. (+10.0)
Assignment 0: Hello World CommandLine!
Testing for grade weight: 10.0
    can you use commandLine?
        Expecting: "Hello World Person!" and found it. (+10.0)
100.0
```