



# Java 1: Your First Program

The Basics of Java



# Getting Started With Java

Once you have the JDK and JRE installed, it's time to get programming.

Our first program is going to be “Hello, World!”. Our *goal* with this program is to have it output the word “Hello, World!” into the **console**.

On the next few slides we will cover the **Source Code**, and how to compile it.



# “Hello, World!” Source Code

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

Note: You cannot just copy and paste this! It's very important to write out the code, it helps you learn and get comfortable with your text editor!



# The Class Declaration

The top line of our source code is a *Class Declaration*. In Java, everything is a part of a Class. You can think of a Class like a container for your code. You can have multiple classes in one file, and multiple classes within other classes. Classes are important for later when you want to get started with Objects (but don't worry about that now!)

```
public class HelloWorld{
```

The Keyword “public” means anything can access this class (as long as it's within its **Scope!** More on that later...) and if not stated explicitly, is implied by default, meaning you don't have to always write public! The Keyword “class” means that this is a class, fairly straight forward. Finally, “HelloWorld” is our class name. The file you save must have this exact name! Otherwise the Java compiler will throw an error.



# The Main Declaration

The code that Java will execute must be within a function called “main”. When run, Java looks for this function to essentially kickstart the execution.

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

We won't go over public again. “Static” is a tricky keyword. Static means that the **Method** isn't apart of an **Instance** of the class, and belongs to the class itself, meaning you don't need to **Instantiate** an Object first. For now, just know that main must always be static. Void is a function return value. Void means that nothing is expected to return from the function. This will be covered more in functions, but for now know that main must always be a void function. Main is our function name, as expected. “String args[]” is our function Parameters. This will also be covered in functions, but for now know that String args[] is our command line arguments.



# The Print Statement

Unfortunately, Java isn't very pretty about its print to console statement.

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

System refers to the Object that represents out console. "out" is a property of that object, and finally "println" is a method that handles outputting to console. Within that we can put a String, and it will output that to the console, with no problems.

Typically in main we would put our code, and all the branching that goes with it



# Saving our code

Java source files are saved using the .java extension. As stated earlier the file name must be the same as the class that has main in it. For our example, our file is called “HelloWorld.java”. Once you have this, you can place it anywhere you want on the computer.

I encourage you to try changing the file name, and seeing what happens after we try to compile.



# Compiling our code

In the same place you have HelloWorld.java, hold down Left Shift and Right Click anywhere in the place you have the file. The menu that appears should have an option that says “Open PowerShell window here”. Click that option and a new PowerShell console will appear.

Now we have to compile our code. To compile it we use the Java Compiler “javac” in our command line. Type “javac HelloWorld.java” to compile it. If it worked, you will see a new file called “HelloWorld.class”.

If you encountered an error, see the next slide, otherwise we can continue on to running our program.





# Errors Compiling

The most common error you will face is that javac **is not recognized as the name of a cmdlet, function, script file, or operable program**. All this means is that the Environmental Variables for java have not been set up. To fix this do this: search your PC for environment variables. Then, click Environment Variables on the bottom of the screen. In “System Variables” locate “Path” if you do not see “Path” click “New” and make “Path”. Both steps are the same now, click “Browse Directory” or “Browse...” and find your Java install. The default location is “C:\Program Files\Java\jdk-16.0.2\bin”. The jdk- will be different based on which version you have installed. Select that path and restart PowerShell and try again, it should now work.



# Running the Program

To run a program in PowerShell, we use the Keyword “java” followed by the Class File. For our example, we would write into PowerShell: “java HelloWorld”. Notice how we exclude the extension. Try running the program with the extension, and you’ll notice the issue; Java cannot find the main function since we didn’t give it the correct class name!

Try changing the file name and class name to experiment.



# Your First Program, Complete

Congratulations! You have now made your first program. It is encouraged to experiment with this code. Try changing “Hello, World!” to something else. We will use this code again in part 2, where we will cover basic variables.



# Definitions

Source Code - The code you write that will be compiled.

Scope - The container of a function which defines what is within it.

Method - A function that is associated with a specific class.

Instance - An Object that is related to a class