

# Lesson 1.2

## *Java Environment Setup*

By the end of this lesson you will learn:

- Things we need to code in Java
- How to check whether Java Environment is already exists in your Computer
- How to install JDK
- How to configure path variable
- How to create a Java Project for any Text Editor
- How to compile a Java Program
- How to Run a Java Program

---

In order to code in Java, you must have setup the java programming environment. It includes the tools to write a java program, to compile a java program and to run a java program. Although there are several IDE (Integrated Development Environment) available in the market such as: NetBeans, Eclipse, IntelliJ, JCreator etc., it is highly unlikely that we will be using these in this course. Using these IDEs are simple, just like you used Visual Studio or CodeBlocks for C++. For this course, we want you to learn everything from the scratch. That's why we will be writing our codes in a text editor and will be giving commands in console to compile and run our programs.

### **Text Editor**

The first thing we need is a Text Editor. A simple text editor comes along with any operating system and we can use that text editor to code. However, coding in plain black and white surface becomes quite dull and boring over time. So, any text editor that can highlight different keywords, constants and statements in a colorful manner, is highly preferable. Lots of text editor are capable of serving our purpose. Among those, Notepad++ and Sublime Text Editor are widely used. You can download any of these two from the following links:

Notepad++ download link : <https://notepad-plus-plus.org/downloads/>  
Sublime Text Editor download link : <https://www.sublimetext.com/3>

After downloading any of the text editor you need to install it. Installation process is exactly like any software installation process. Just follow the steps on your screen and you will be fine.

## Java Development Kit

Now that we have a text editor to write our code, we need something to compile and run it. We need JDK (Java Development Kit). The latest version of JDK as of May 2020 is JDK 14. However, we are preferring JDK 8 or JDK 11. You can download JDK from the following links:

JDK Download Links:   [www.java.com/en/download/manual.jsp](http://www.java.com/en/download/manual.jsp)  
                                  [www.oracle.com/java/technologies/javase-downloads.html](http://www.oracle.com/java/technologies/javase-downloads.html)

This JDK comprises of Java Compiler that compiles java source code and other development tools along with JRE (Java Runtime Environment). JRE consists of JVM (Java Virtual Machine) and Java Runtime Library Classes. Inside JVM there is JIT (Just-in-Time) compiler. The Java Compiler generates bytecode after compiling the java source codes. This bytecode is platform independent and so, it is same across all platforms. JRE provides the environment to run the bytecode. JVM is platform depended. Different JVMs for different platforms interprets the same bytecode and generate Machine Readable code. JIT compiler executes this machine readable code in real time piece by piece.

The JRE component can also be downloaded separately. If you install JDK in your computer, JRE gets installed in the process. As a result, you can both compile and run java code in your computer. On the other hand, if you install JRE, you will not be able to compile java codes but you can surely run the bytecode which was generated after the compilation of java source code. It is only possible as the bytecode is same across all the platforms. So, bytecode generated in one platform can run in any platform if the associated platform has a JRE installed in it. As in this course, we need to write source codes and compile them, we should download and install JDK.

## Installing JDK and Configuring Environment Variable

Till now, we have downloaded and installed a text editor for writing java codes and we have downloaded JDK. Before installing JDK in our PC, first we should check whether it is already installed or not. The following steps illustrate the process:

- Step 1: Go to **Start**
- Step 2: Type **cmd** and open **Command Prompt**.
- Step 3: Give a command **java** and hit the **Enter** key.

After Step 3, there will be any of the following two cases:



Case 1: Not Installed



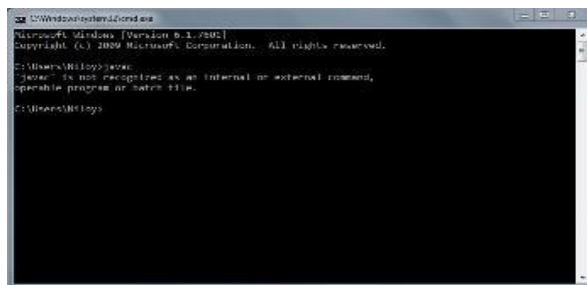
Case 2: Installed

Step 4: If ***“Not Installed”***, Install ***JDK*** and go to 5. Else go to 5.

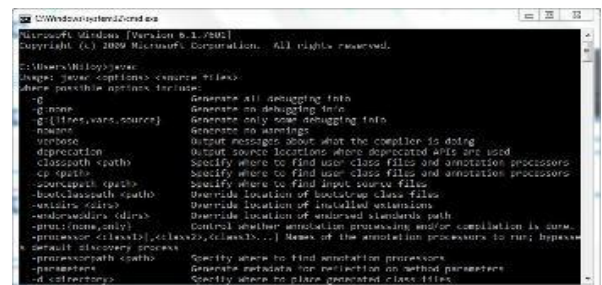
Step 5: Open ***Command Prompt*** again.

Step 6: Give a command ***javac*** and hit the ***Enter*** key.

After Step 6, there will be any of the following two cases:



Case 1: Path Not Configured



Case 2: Path Configured

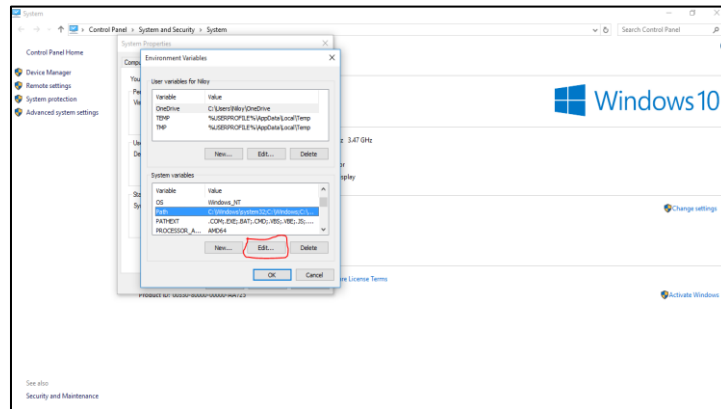
Step 7: Configure Environment Variable

- (a) Go to ***This PC***
- (b) Go to ***C Drive***
- (c) Go to ***Program Files***
- (d) Go to ***Java***
- (e) Go to ***jdk1.8.xxxx***  
(xxxx is the version number)
- (f) Copy the path specified in the picture

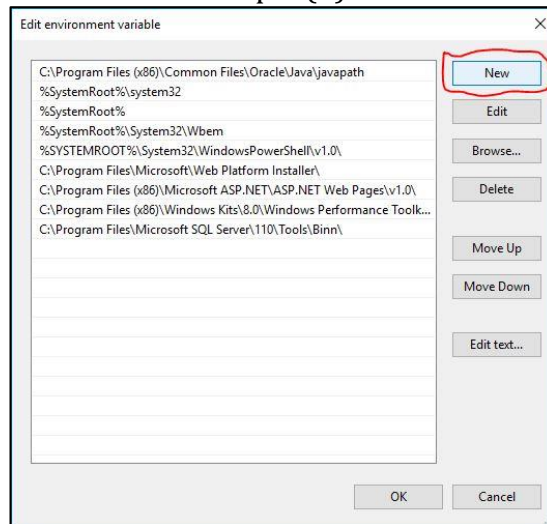
Step 8: Now to paste the copied path:

- (a) Right Click on ***This PC*** and select ***Properties***
- (b) Click on ***Advanced System Settings***
- (c) Select ***Environment Variable***
- (d) Select ***Path*** from the bottom box and click on ***Edit***
- (e) Click on ***New***
- (f) Paste the path and click on ***OK***.

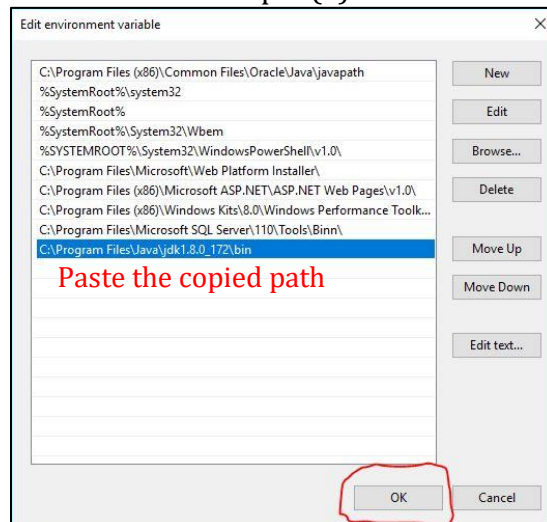




Step 8 (d)



Step 8 (e)



Step 8 (f)

Step 11: Open **Command Prompt** again and give the **java** command again.

Step 12: Give the **javac** command again.

The following two cases should appear respectively:



Step 11



Step 12

## Creating a Java Project

After we have completed installing JDK and configuring environment path variable, we will now create our 1<sup>st</sup> Java project. We can follow the following steps:

**Step 1:** Go to Your **preferred directory** (Any drive apart from **C Drive**) and Create a **folder**. We will create all the projects inside this folder. You may name this folder as per your wish. Let us assume that the name of this folder is **"Java Projects"**.

**Step 2:** We will be creating separate folders for each of our projects. So, inside the **Java Projects** folder, create another folder. The name of this folder should indicate what project we are doing inside it. For Example: we are going to do a **"Hello World"** application as our 1<sup>st</sup> project. So, let's name this folder **"HelloWorldApp"**.

**Step 3:** Inside the **HelloWorldApp** folder we will create a text file. We will have to rename this file as per the name of the class we will be writing in it. It is because of the coding convention of writing one class inside one file. Let's assume that the name of the class will be **Hello**. So, we have to rename the file as **Hello** and we also need to change the extension of file from **.txt** to **.java**.

We have now successfully completed creating our 1<sup>st</sup> project in Java. All that remains is just to write a code in the **Hello.java** and start our journey of exploring the magic of Java.

## The Hello World Program

The following is the Hello World Program that we will be writing inside the **Hello.java** file:

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

## Compiling and Running a Java Project

After writing the above code in the ***Hello.java*** file, we need to compile it. But, before compiling we have to save the file. If we do not save the file, unsaved file with partial code or even in some cases an empty file with no code will be compiled.

As we stated earlier that, we will be giving commands in command prompt to compile and run our code, first we need to open command prompt in the directory we have saved our code (in this case, in the HelloWorldApp folder).

After opening the command prompt, we have to use the ***javac*** command to compile the ***Hello.java*** file. We need to write the following and hit the ***Enter*** key:

```
javac Hello.java
```

If there is any error in the code, it will be shown in the command prompt. If there is no error in the code, the ***Hello.java*** file will be compiled successfully and a ***Hello.class*** file will be generated inside the ***HelloWorldApp*** folder. Now, to run the code we have to use the ***java*** command and specify the name of the class that contains the main method (in this case, the class name is ***Hello***) in the command prompt. So, we need to write the following and hit the ***Enter*** key:

```
java Hello
```

The code will run and will display the sentence “***Hello World***” as output.

## Practice

- ✓ Create a Project named ***Test*** in java that has one class named ***Test*** in it. The Test class contains the main method and the main method prints the sentence “***This is a Test***”.