Compiling Your First Java Program!

In this lesson, an explanation of how to compile your code in Java, which compilers to use and the most suitable ones for different Operating Systems, is provided.

We'll cover the following

For the computer to execute the *Hello World* code you have written, it first needs to be compiled by a Java compiler. The compiler translates the computer code written in a particular programming language into the machine code.

What does the compiler do?

In very broad terms:

> **Did you Know?**
>
> The compiler is a translator that acts as an intermediary between the programmer and the CPU on the computer.

A high-level language like Java is actually a sort of *'compromise'* language between the native language of the CPU (generally referred to as machine language) and the native language of the programmer (say English).

Computers do not natively understand human languages. Therefore, the compiler is reading the code written by the programmer and translating it into machine language code that the computer can execute directly.

In `Java`, programs are **not** compiled into **executable files**; they are compiled into *bytecode*, which the JVM (Java Virtual Machine) then executes at runtime. Java

source code is compiled into *bytecode* when we use the javac compiler. The *bytecode* gets saved on the disk with the file extension `.class`. When the program is to be run, the bytecode is converted using the just-in-time (JIT) compiler. The result is machine code, which is then fed to the memory and is executed.

In short, we follow the following two steps to execute the Java program:

1. Java programs need to be compiled to **bytecode**.
2. Then we need to convert the **bytecode** into **machine code**.

The Java classes/bytecode are compiled to machine code and loaded into memory by the JVM when needed the first time. This is different from other languages like C/C++, where programs are to be compiled to machine code and linked to create an executable file before it can be executed.

Running the compiler

The code needs to be compiled with a compiler to finish the process.

What if you don't have one?

Well, the good news is, there are *several* good compilers that are available for free.

- The Eclipse has various versions available for most systems and does a good job of implementing the **Java** standard libraries.
- The NetBeans has complete support for Java used for multiple purposes, including **WebApps** and **Web-Services**. It also provides Git Repositories to its users.
- Some Mac Users prefer using Xcode for running Java on their machine.

There are multiple compilers for one language, and picking out one is totally a personal preference.

Compilation procedure

We are providing a Java execution environment in our course, so you don't have to worry about this. But, if you prefer to install the environment on your own machine for other uses, you can follow the instructions given in this link.

We hope you've successfully installed and run your first Java program.

That's it for now, let's play around with some print statements in the next lesson to get *'you'* used to it.