Compiler: A compiler is a software program that translates the entire source code of a program at once into machine code (binary code that the processor can understand)**.**

If there are any errors in the code, the compiler will show all the errors together after checking the whole program.

Interpreter: An interpreter is a software program that translates and runs the code line-by-line.

If there is an error, it **stops at that line** and shows the error. The rest of the code will not run until the error is fixed.

How C & C++ works:

In **C and C++**, we write a program in the form of source code.

The **compiler** translates this source code **directly into machine code** Then, the **processor executes the machine code**, and the program runs.

How Java works:

The Java **compiler** translates this source code into an intermediate form called **bytecode**.

Bytecode is **not directly understandable by the processor**, so it cannot be executed, the **JVM (Java Virtual Machine)**, which acts as an **interpreter**, reads the bytecode and **converts it into machine code** that can be understand by processor, and the program runs.

Why C & C++ were lacking? What is the exact problem java is trying to solve?

C and C++ are powerful languages but have major drawbacks like platform dependency which is why Java was introduced to overcome them with features like JVM and platform independence.

Java was designed to solve platform dependency issues found in C/C++. It allows developers to write code once and run it anywhere.

Bytecode: Bytecode is an intermediate code generated by the Java compiler after compiling, It is not machine code, but a special code that JVM (Java Virtual Machine) can understand and execute.

Key Points:

It is platform independent.

It cannot run directly on the processor.

The JVM reads bytecode and converts it to machine code for that specific system.

Platform Dependent:

Definition: If a program can run only on the same OS where it was compiled, it is called platform dependent.

Example: C/C++

When you compile a C program on Windows → the compile code is for Windows only. If you take that compile code to Linux → it won't work.

Because the compiled code is specific to the OS and processor where it was compiled.

Platform Independent:

Definition: If a program can run on any OS, it is called platform independent.

Example: Java

It can be executed on any platform that has JVM (Java Virtual Machine).

JVM is available for Windows, Linux, Mac. So same code can run on all platforms.

How to install java?

To install Java, first I downloaded the latest JDK from the Oracle website based on my operating system. then ran the installer and completed the installation. After that, I have set the PATH environment variable by adding the JDK's bin folder path. Finally, I verified the installation using java --version and javac commands in the command prompt.

What is PATH Environment Variable?

PATH is a special system environment variable in your computer's operating system that tells the system where to find executable files like java, javac etc.

after Java installed: copy the path

Ex: C:\Program Files\Java\jdk-21\bin

Inside this bin folder, files like java and javac exist. But your Command Prompt won't know where to find them unless you add this path to the PATH variable. ---

The PATH environment variable is used to specify the directories in which executable programs are located. In Java, after installation, we add the bin folder of JDK to the PATH so that we can run commands like java and javac from any location in the system.

Purpose of PATH Environment Variable:

Tells OS where to search: When you type java or javac in command prompt, the system searches for it in the folders listed in PATH.

Run tools from any location: You can compile/run Java programs from any folder without navigating to the JDK bin folder.

Saves time & avoids errors: Without it, you'll get an error like

```
java is not recognized as an internal or external command, operable program or batch file.
```

Internal Command: These commands are already available in the Command Prompt.

External Command: To run these commands, we need to tell the system where the required files are located (by setting the path).

JDK – Java Development Kit

It is a full software development kit to write, compile, and run Java programs

JRE – Java Runtime Environment

It is only for running Java programs. You can't compile programs with JRE.