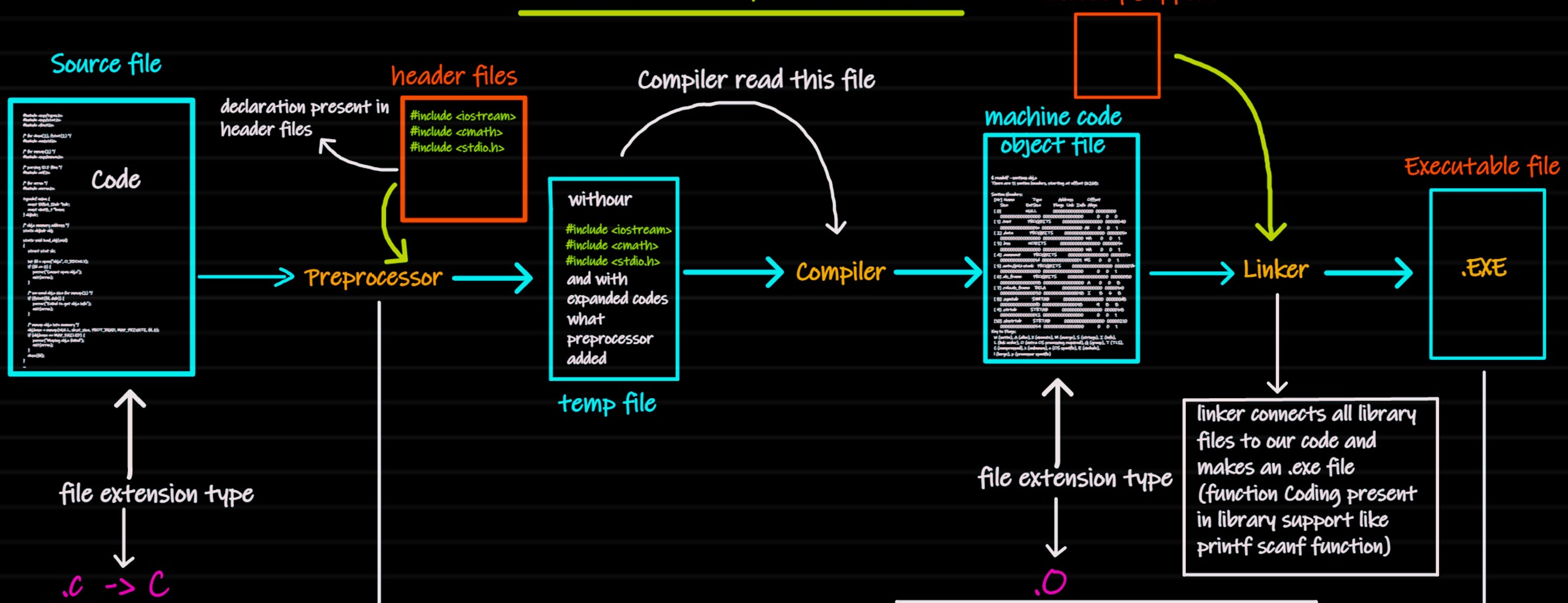
Why Java is platform independent language

Written BY Subham(CodeXam)

Software Development In C/C++



linker connects all library files to our code and makes an .exe file (function Coding present in library support like printf scanf function)

.c -> C

.cpp -> C++

An object file is a computer file containing object code, that is, machine code output of an assembler or compiler. The object code is usually relocatable, and not usually directly executable. There are various formats for object files, and the same machine code can be packaged in different object file formats

Preprocessor expends the source code

It is not a part of a compiler

It is a text substitution tool

```
#include <iostream>
#include <cmath>
#include <stdio.h>
```

-> it is called preprocessor directive / commands

this means we are including separate file in our program

After the preprocessor expands these files, it begins adding those codes to your code.

-> for running this .exe file we need a computer with a operating system(OS)

-> OS + Hardware(H/W) = Platform

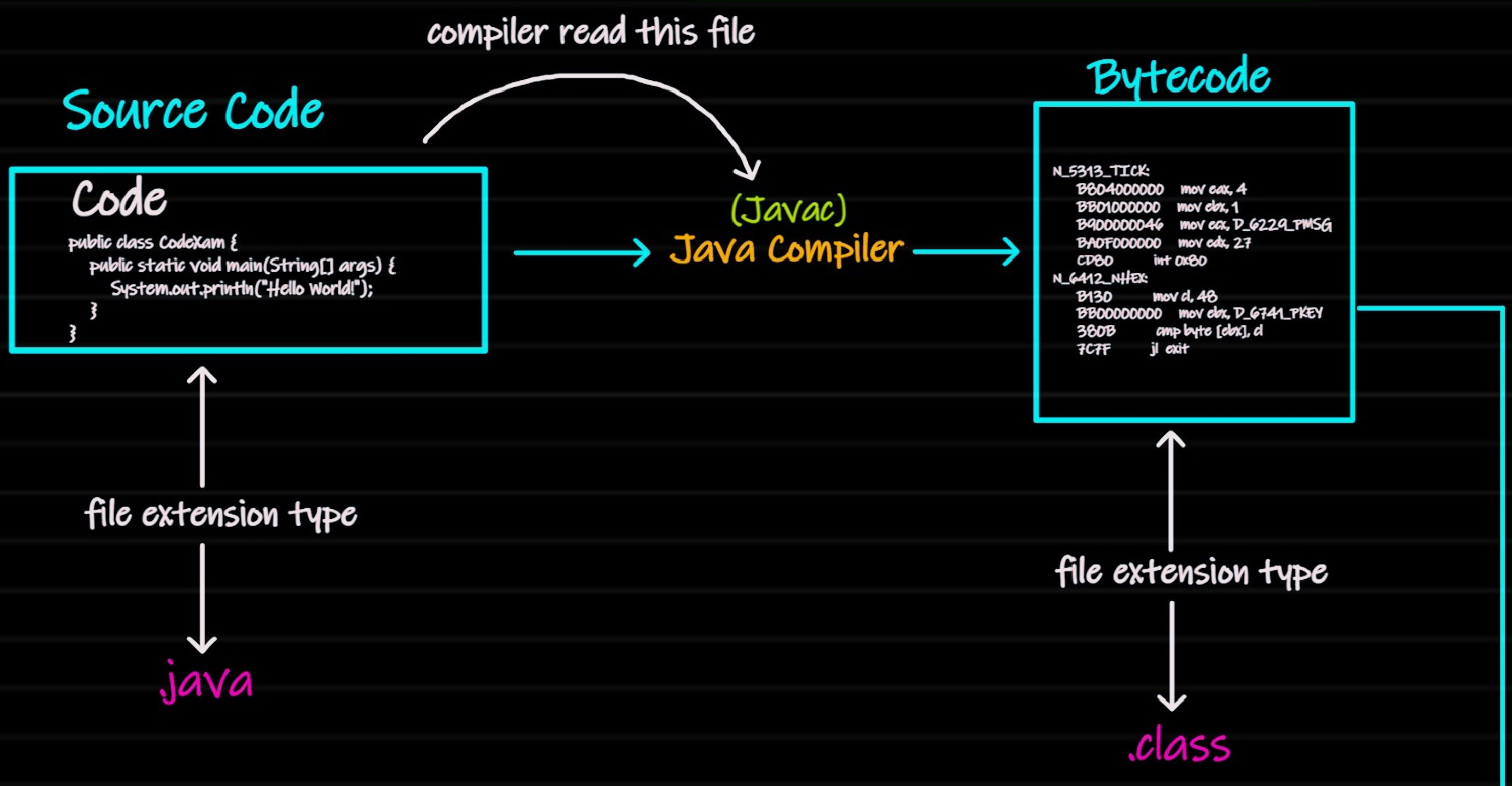
The .exe file can only run on Windows; if we want to run this code on another operating system, we need another compiler it is called platform dependent

Page 2

Why Java is platform independent language

Written BY Subham(CodeXam)

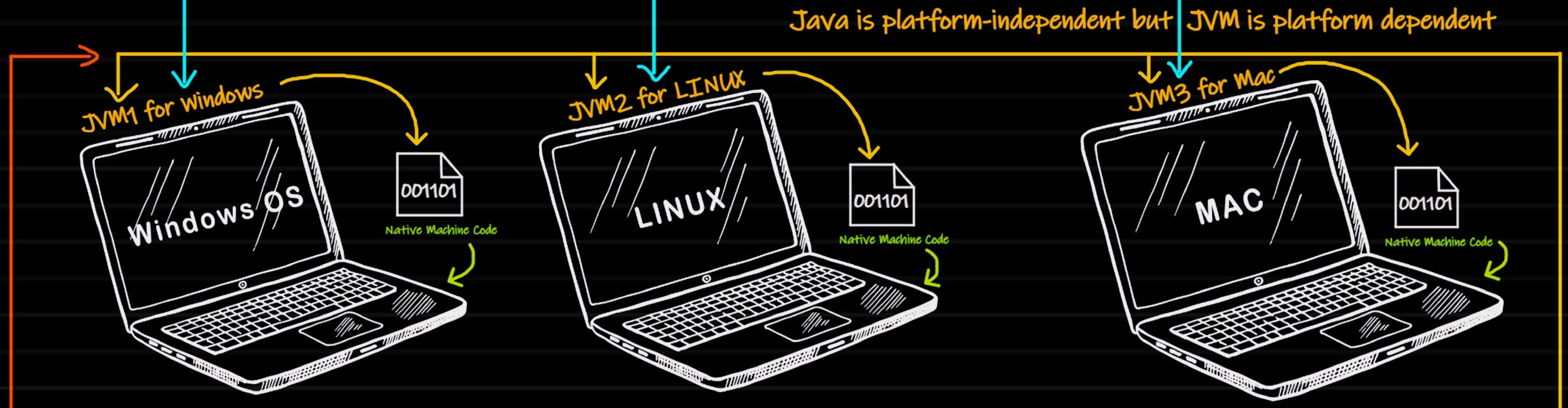
Software Development In JAVA



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

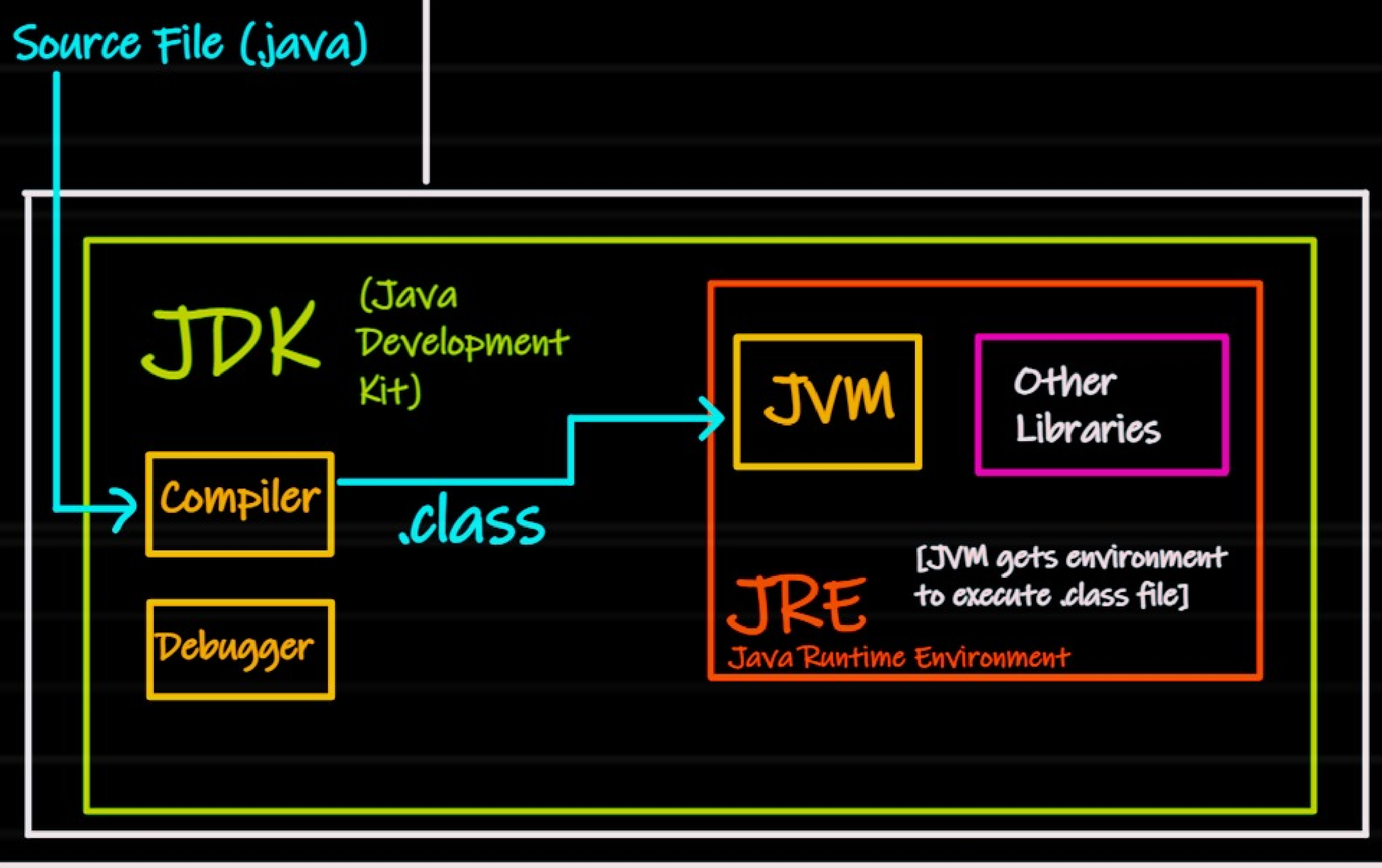
```
N_5313_TICK:
    B804000000   mov eax, 4
    BB01000000   mov ebx, 1
    B900000046   mov ecx, D_6229_PMSG
    BA0F000000   mov edx, 27
    CD80         int 0x80
N_6412_NHEX:
    B130         mov cl, 48
    BB00000000   mov ebx, D_6741_PKEY
    380B         cmp byte [ebx], cl
    7C7F         jl  exit
```

Java is platform-independent but JVM is platform dependent



There are seperate JVMs for each OS. You have to install the JDK as per your OS

OS + Hardware(H/W) + JVM = Platform for Java



[JVM gets environment to execute .class file]

The JDK is a development environment for building applications, applets, and components using the Java programming language. The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

The JIT compiler compiles the bytecodes of that method into native machine code, compiling it "just in time" to run. When a method has been compiled, the JVM calls the compiled code of that method directly instead of interpreting it.

Class Loader

Bytecode Verifier

JIT Compiler

JVM (Java Virtual Machine)

The JVM is what takes the bytecode and translates it into machine code.

Why Java is platform independent language

Written BY Subham(CodeXam)

The Writer's Explainer

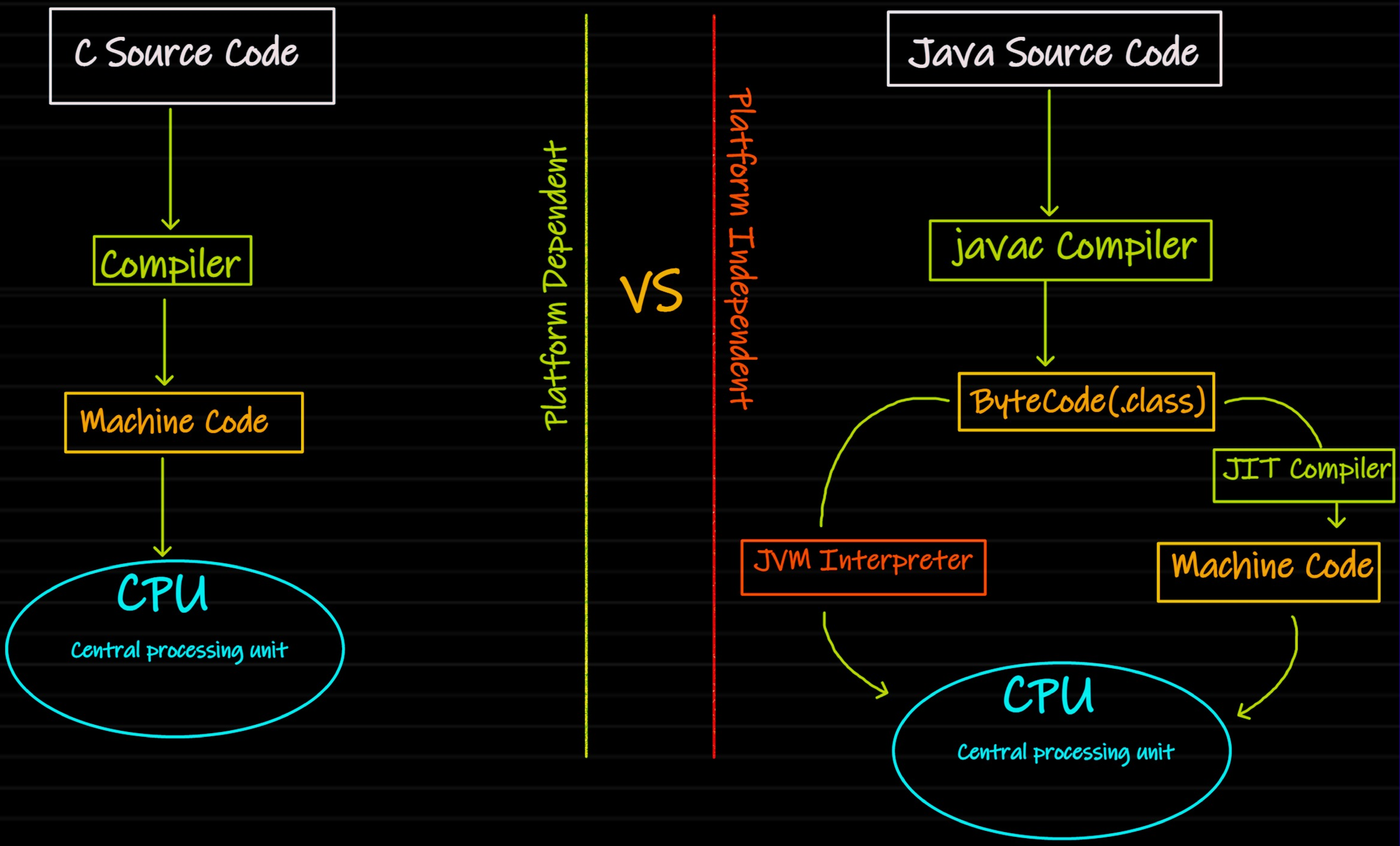
Java compiler produces a unique type of code called bytecode unlike C compiler where compiler produces only natively executable code for a particular machine.

When the Java program runs in a particular machine it is sent to java compiler, which converts this code into intermediate code called bytecode. This bytecode is sent to Java virtual machine (JVM) which resides in the RAM of any operating system. JVM recognizes the platform it is on and converts the bytecodes into native machine code. Hence java is called platform independent language.

Java is platform-independent but JVM is platform dependent why?

This fact can be verified by trying to download the JVM for your particular machine – when trying to download it, you will be given a list of JVMs corresponding to different operating systems, and you will obviously pick whichever JVM is targeted for the operating system that you are running. So we can conclude that JVM is platform-dependent and it is the reason why Java is able to become "Platform Independent".

Basically, Java doesn't require the entire code to be rewritten for all the different platforms. It supports platform independence by using Java byte-code and Java Virtual Machine. Java compiler javac converts the program code into byte code, this byte code is platform-independent and can be run on any operating system's JVM. JVM interprets the byte code to machine code and the program is executed.


C Source Code
Compiler
Machine Code
CPU
Central processing unit
Platform Dependent
VS
Platform Independent
Java Source Code
javac Compiler
ByteCode(.class)
JIT Compiler
JVM Interpreter
Machine Code
CPU
Central processing unit