Java Course By CodeWithHarry

Java is an Object Oriented programming language developed by Sum Microsystems of USA in 1991
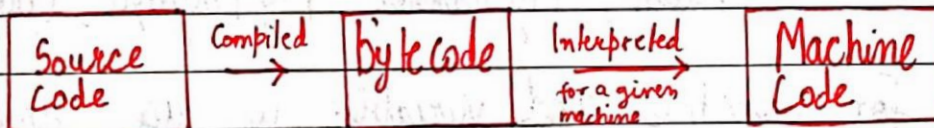
It was originally called Oak by James Goslin
   ↳ one of the inventors of Java!

JAVA = Purely object Oriented

How JAVA Works?
Java is Compiled into the bytecode and then it is interpreted to machine code

| Source Code | Compiled → | byte code | Interpreted → for a given machine | Machine Code |

JAVA Installation
Go to Google & type "Install JDK" ⟹ Installs JAVA JDK
Go to Google & type "Install IntelliJ Idea" ⟹ Installs JAVA IDE

JDK → JAVA Development Kit = Collection of tools used for developing and running Java programs

JRE → JAVA Runtime Environment = Helps in executing programs developed in JAVA

BYTE-CODE: It's a low-level, platform-independent representation of the source code that's easier for the interpreter or virtual machine to understand and execute efficiently.

Public In Java, the main method must be declared as public so that the Java Virtual Machine (JVM) can access and execute it.

static: This keyword indicates that the main method belongs to the class itself, rather than any specific instance of the class.

void: This is the return type of the main method, indicating that the method does not return any value.

main This is the name of the method. In Java, the main method serves as the entry point for executing a Java program

(String[] args). This is the parameter list of the main method. In Java, the main method can accept an array of strings as command-line arguments. The args parameter allows you to pass arguments to the Java program when you run it from the command line

Basic Structure of a Java Program

```
package com.company;        → Groups classes!
                                    Entrypoint into the application
public class Main {
    public static void main (String [] args) {
        System.out.println ("Hello World");
    }
}
```

Naming Conventions

→ For classes, we use PascalConvention. First and subsequent characters from a word are capital letters (uppercase)

Example:
  Main, My Scanner, My Employee, CodeWithHarry

→ For functions and variables, we use CamelCase Convention. Here first character is lowercase and the subsequent characters are uppercase like below:
  main, myScanner, myMarks, CodeWithHarry

Working of Byte code:

Here's how bytecode works in a nutshell:

Compilation: When you write code in a high-level programming language like Java or Python, it's first translated into bytecode by a compiler. This bytecode is not machine code that directly runs on the hardware but rather a set of instructions that the interpreter or virtual machine can understand.

Execution: The bytecode is then executed by a runtime environment like the Java Virtual Machine (JVM) for Java or the Python interpreter for Python. The interpreter reads and interprets each bytecode instruction, carrying out the corresponding actions in a way that's optimized for the specific runtime environment.

Platform Independence: One of the key advantages of using bytecode is that it enables platform independence. The same bytecode can be executed on different operating systems or architectures as long as there's a compatible interpreter or virtual machine for that platform.

Security and Portability: Bytecode can also provide a level of security, as the bytecode is often harder to reverse-engineer compared to the original source code. Additionally, distributing bytecode allows developers to share their software without exposing the source code, which can be important for commercial or proprietary software.