Java: A Perfect Blend of Compilation & Interpretation! 🚀

Ever wondered how Java runs seamlessly across different platforms? Let's break it down in a simple and structured way!

🔷 Key Components of Java Execution

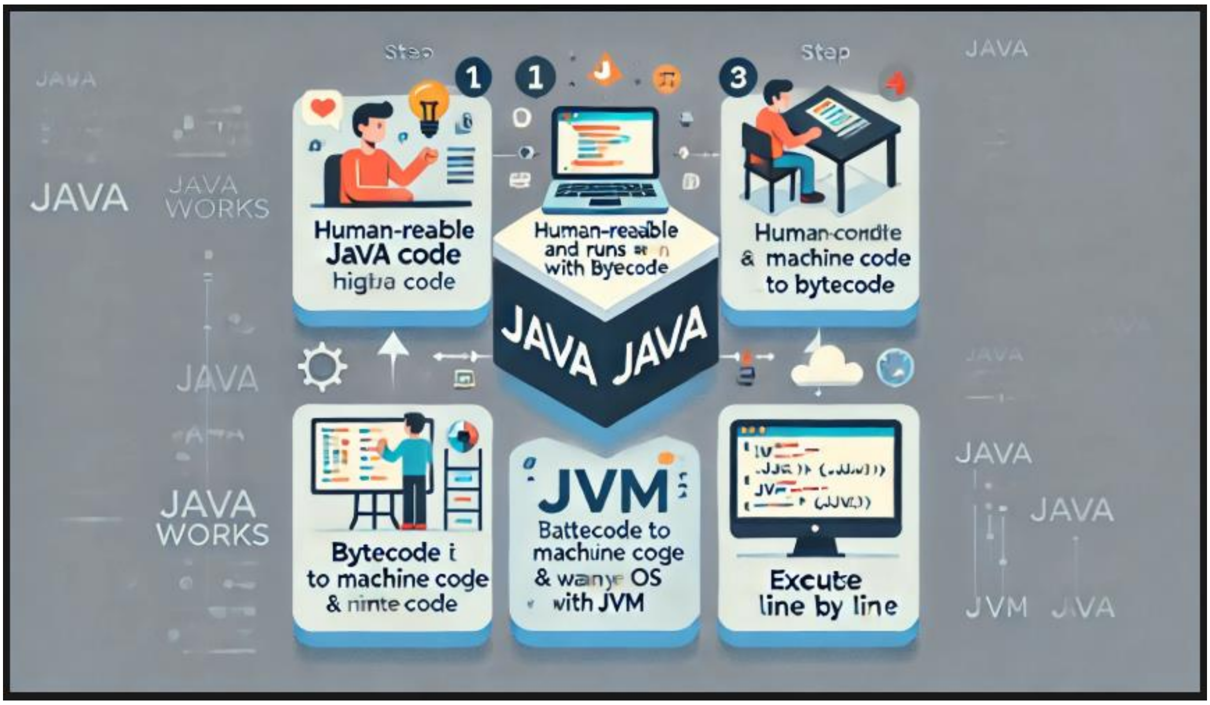
- **Source Code** → The original Java code, written in a human-readable format.
- **Bytecode** → The compiled Java code (.class file), which is platform-independent and runs on any OS with a **Java Virtual Machine (JVM)**.
- **Native Code** → For system-level access, Java uses **Java Native Interface (JNI)** to call C/C++ code when needed.

🔄 How Java Works

1️⃣**Writing the Code** – Developers write Java programs using a high-level language.
2️⃣**Compilation** – The **Javac Compiler** converts filename.java into filename.class (Bytecode).
3️⃣**Platform Independence** – The **JVM** ensures that the same Bytecode runs on any operating system (Windows, Linux, Mac, etc.).
4️⃣**Bytecode Execution** – The **JVM** translates Bytecode into **Native Machine Code (Low-Level Language)**.
5️⃣**Interpretation & Output** – The JVM **interprets and executes** the code line by line, generating the final output.

✅ Write Once, Run Anywhere!

This is the true power of **Java's architecture**—providing flexibility, efficiency, and cross-platform compatibility.


Step
1
1
3
Step
JAVA
JAVA
JAVA
JAVA WORKS
Human-reable JaVA code higba code
Human-reaable and runs with Byecode
Human-condte & machine code to bytecode
JAVA JAVA
JAVA
JAVA
Bytecode i to machine code & ninte code
JVM
Battecode to machine coge & wanye OS with JVM
Excute line by line
JAVA WORKS
JAVA
JAVA
JVM JAVA