🔁 Step-by-Step Sequence of Java Compilation

✅ 1. Write Java Code

You write your code in a file with .java extension (e.g., Hello.java).

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

⚙️ 2. Compile the Code using javac

The Java Compiler (javac) converts .java file into a .class file.

```
javac Hello.java
```

✅ This creates Hello.class — it contains **Bytecode**, not machine code.

📦 3. Bytecode is Platform-Independent

- The .class file contains **Bytecode**, which is the **intermediate code**.
- Bytecode is **not specific to any OS or hardware** — it's portable.

🔄 4. JVM Executes the Bytecode

You run the code using the **Java Virtual Machine (JVM)**:

```
java Hello
```

JVM reads the Bytecode from Hello.class and:

- Uses an **Interpreter** or **Just-In-Time (JIT) Compiler**
- Converts Bytecode into **machine code**
- Runs it on your system

🔍 In Short:

Step	Action	Tool Used	Output
1	Write code	Text Editor / IDE	Hello.java
2	Compile code	javac	Hello.class (Bytecode)
3	Run code	java command (JVM)	Executes program

🔑 Key Concepts:

- **JDK**: Java Development Kit (includes compiler javac, tools, and JVM)
- **JVM**: Java Virtual Machine (runs the Bytecode)
- **Bytecode**: Intermediate code between source code and machine code