Software Engineering -2

OOP with Java

Nimesha Hewawasam
nimesha.h@nsbm.ac.lk

Why JAVA?

A **high-level language** that can be characterized by all of the following:

- Object-orientated programming language
- Platform independent
- Strongly-typed programming language
- Interpreted and compiled language
- Automatic memory management

Dream Jobs from JAVA

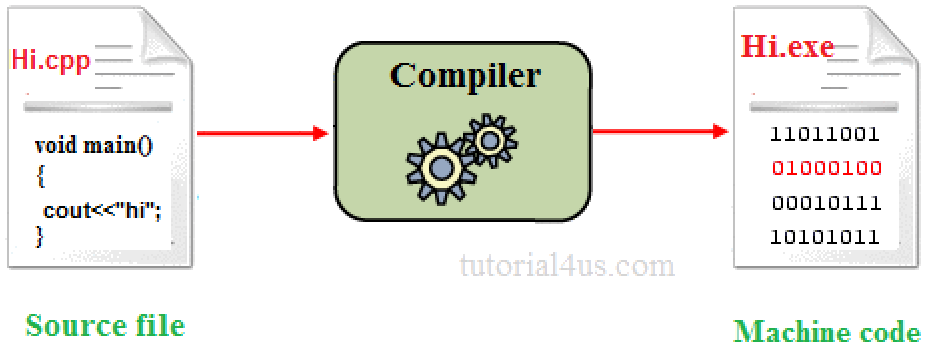
Company	**Focus Area**
IFS Sri Lanka	ERP software — Java-based solutions
WSO2	Open-source integration — Java, Ballerina
99X Technology	Java-based full-stack enterprise software
Sysco LABS	Java for cloud-native apps and microservices
Virtusa	Global delivery with Java-based enterprise solutions
MillenniumIT ESP	Financial platforms, Java backend systems
Zone24x7	Java for IoT, logistics, and cloud platforms
Mitchell Wiggins, CodeGen, Cambio	Healthcare, travel tech, fintech – Java based platforms

Part I

Installation and behind the seen

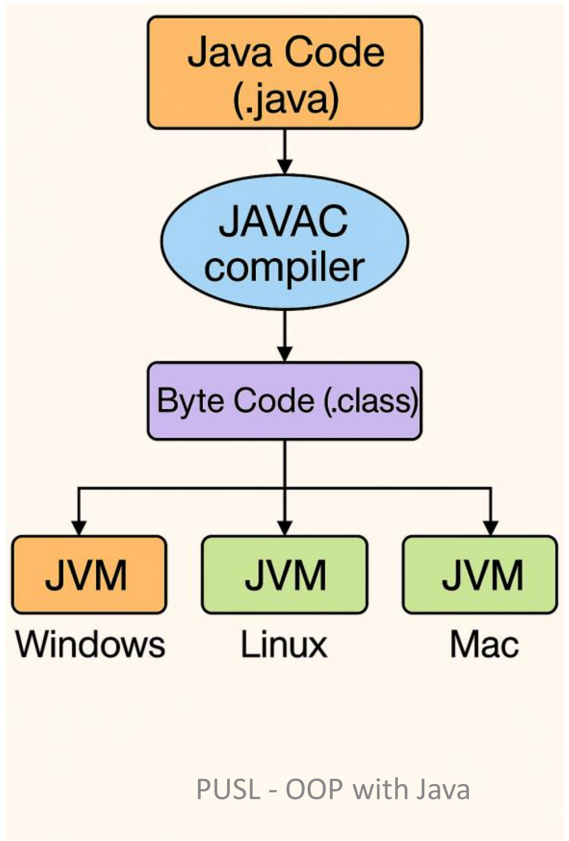
Why we need programming language?

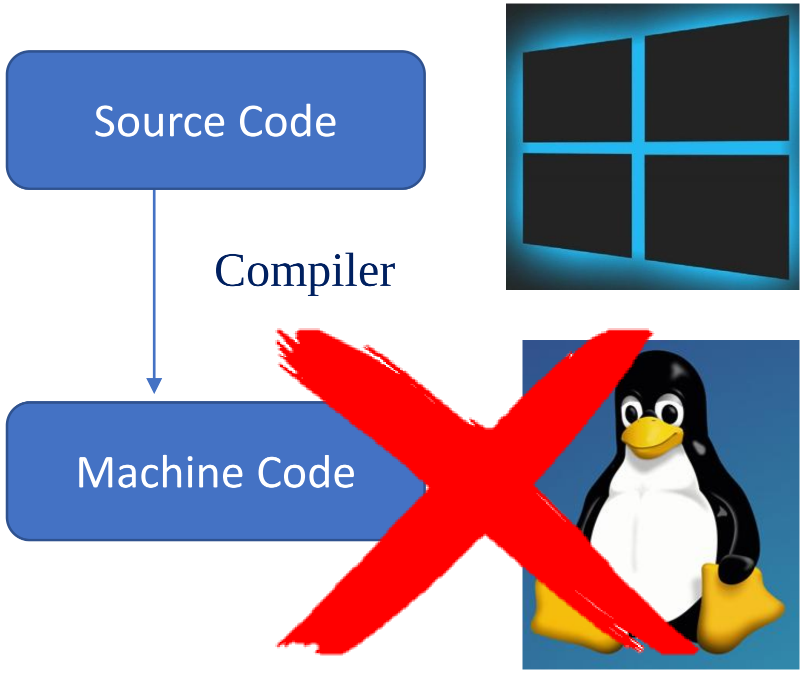
- Computer only understand binary values. 1 and 0
- Through a programming language we provide command to the machine and expecting output.
- Source code : provide command using understandable syntax
- Machine code : way that machine understand
- Compiler: transform source code to machine code.



Why we need programming language?

- We can’t run the compiled code in every OS.
- There for in java we have JVM:
  - Java Virtual Machine





JDK (Java Development Kit)

What it is: A full package to develop and run Java programs.

Contains:

- **JRE :** (Java Runtime Environment)
- **JVM :** (Java Virtual Machine)
- **Development tools**: java, javac (compiler), Javadoc, jar, etc

Used by: Java developers to writing and compiling Java code

JRE (Java Runtime Environment)

What it is: A package to run Java applications (but not develop).

Contains:

- JVM
- Libraries and classes required to run Java programs.

Used by: End users who want to run Java apps without developing them.

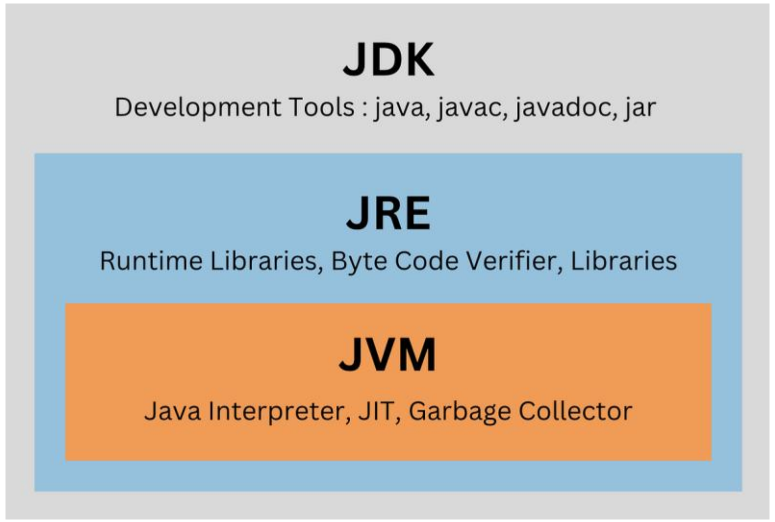
JVM (Java Virtual Machine)

What it is: An abstract machine that executes Java bytecode (.class files).

Platform-specific: There are different JVMs for Windows, Linux, Mac, etc.

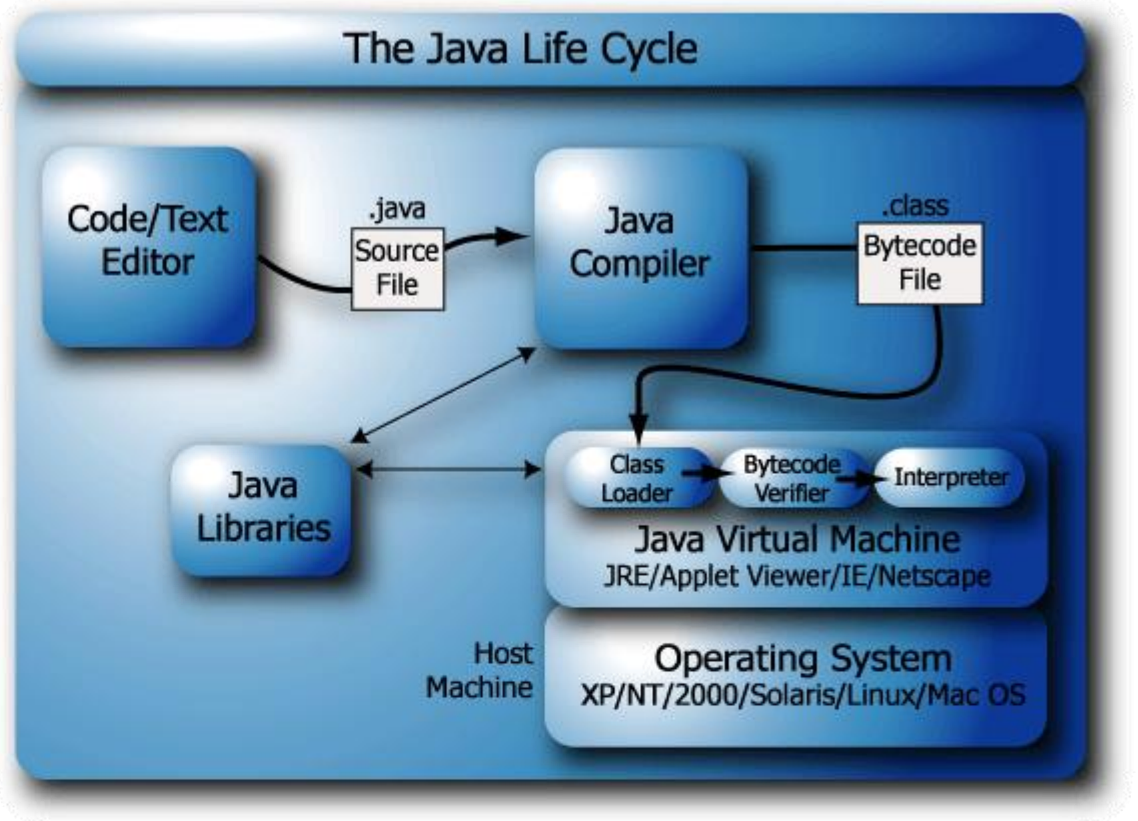
Key features:

- Converts bytecode to machine code.
- Provides platform independence ("Write Once, Run Anywhere")


JDK
Development Tools : java, javac, javadoc, jar
JRE
Runtime Libraries, Byte Code Verifier, Libraries
JVM
Java Interpreter, JIT, Garbage Collector

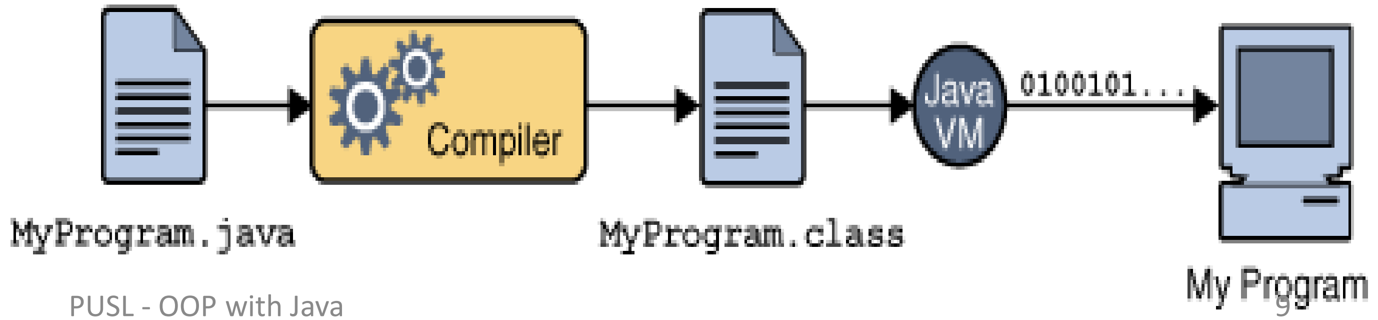
Java life cycle

- **Code editor**:
  - Programmer **writes** program and stores on disk
- **Compile**
  - Compiler creates ***bytecodes*** from program **(.class)**
- **Load**
  - Class **loader** stores bytecodes in memory
- **Execute**
  - **Interpreter**: translates ***bytecodes*** into **machine language**



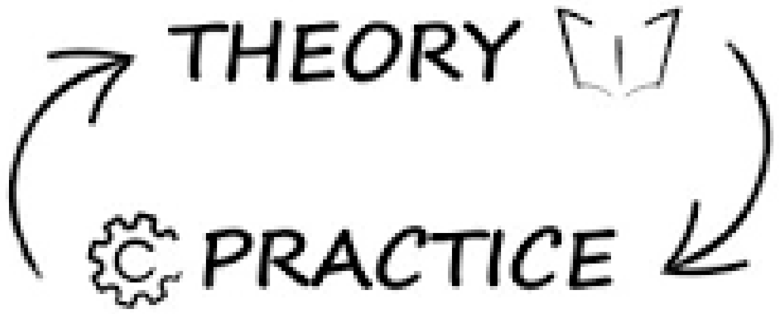
Big Picture

- In the Java programming language, all source code is first written in plain text files ending with the **FileName.java** extension.
- Those source files are then compiled into .class files by the **javac compiler**.
- **FileName .class** file does not contain code that is native to your processor; it instead contains ***bytecodes*** — the machine language of the Java Virtual Machine (Java VM).
- The java launcher tool then runs your application with an instance of the Java Virtual Machine.



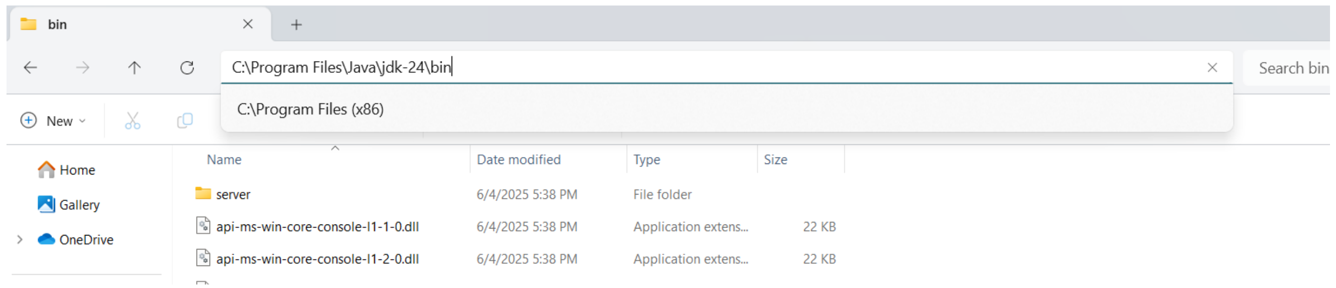
Part II

Practical before Theory



Java Installation

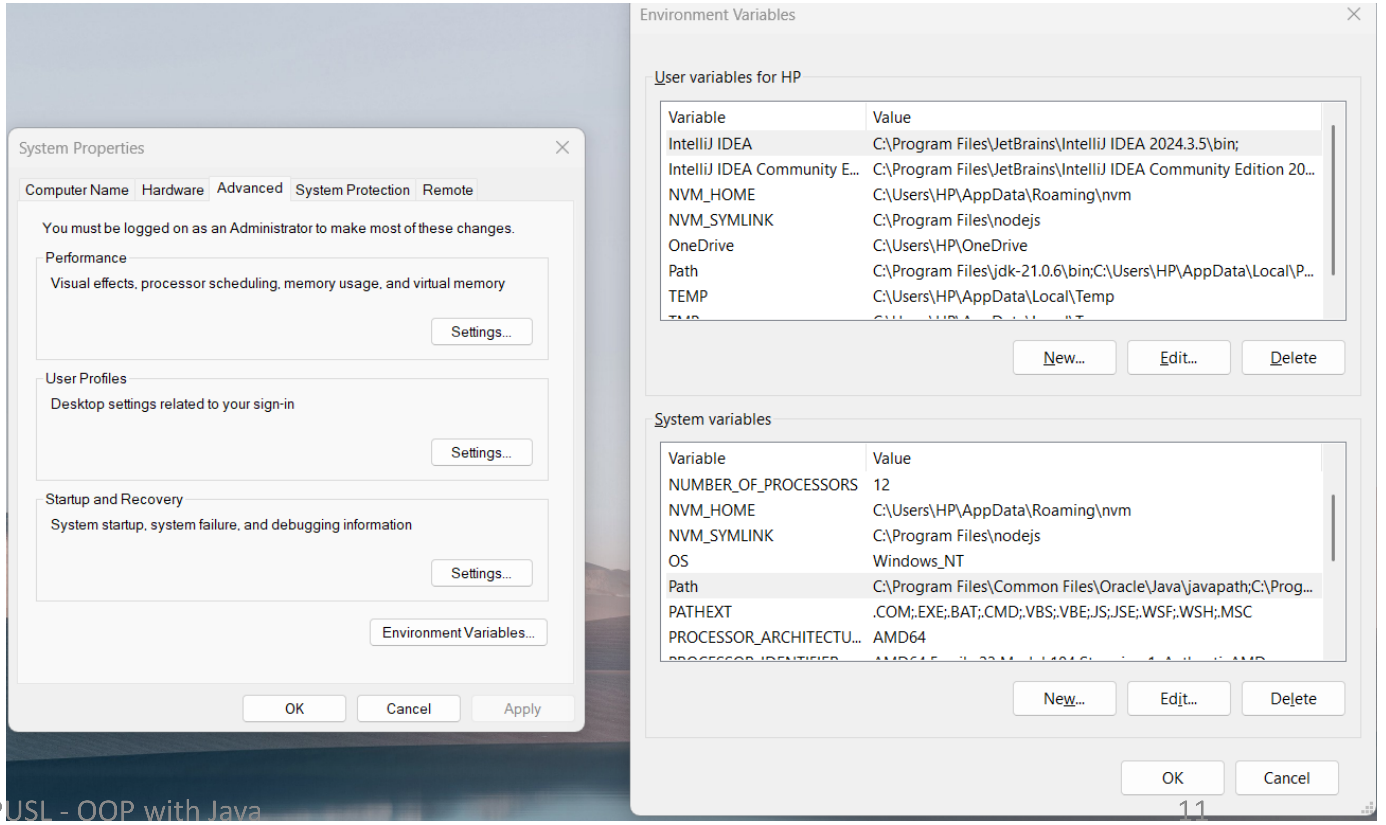
- jdk (java development kit)
  - [Java Downloads]
  - Development Environments
    - [NetBeans]
    - [Intellij]



C:\Program Files\Java\jdk-24\bin

C:\Program Files (x86)

Name	Date modified	Type	Size
server	6/4/2025 5:38 PM	File folder	
api-ms-win-core-console-l1-1-0.dll	6/4/2025 5:38 PM	Application extens...	22 KB
api-ms-win-core-console-l1-2-0.dll	6/4/2025 5:38 PM	Application extens...	22 KB



You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

User variables for HP

Variable	Value
IntelliJ IDEA	C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.5\bin;
IntelliJ IDEA Community E...	C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 20...
NVM_HOME	C:\Users\HP\AppData\Roaming\nvm
NVM_SYMLINK	C:\Program Files\nodejs
OneDrive	C:\Users\HP\OneDrive
Path	C:\Program Files\jdk-21.0.6\bin;C:\Users\HP\AppData\Local\P...
TEMP	C:\Users\HP\AppData\Local\Temp

System variables

Variable	Value
NUMBER_OF_PROCESSORS	12
NVM_HOME	C:\Users\HP\AppData\Roaming\nvm
NVM_SYMLINK	C:\Program Files\nodejs
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Prog...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTU...	AMD64

Compile and Execute Your First Program

1. First Save your file with .java extension in known location .

   HelloWorldApp.java

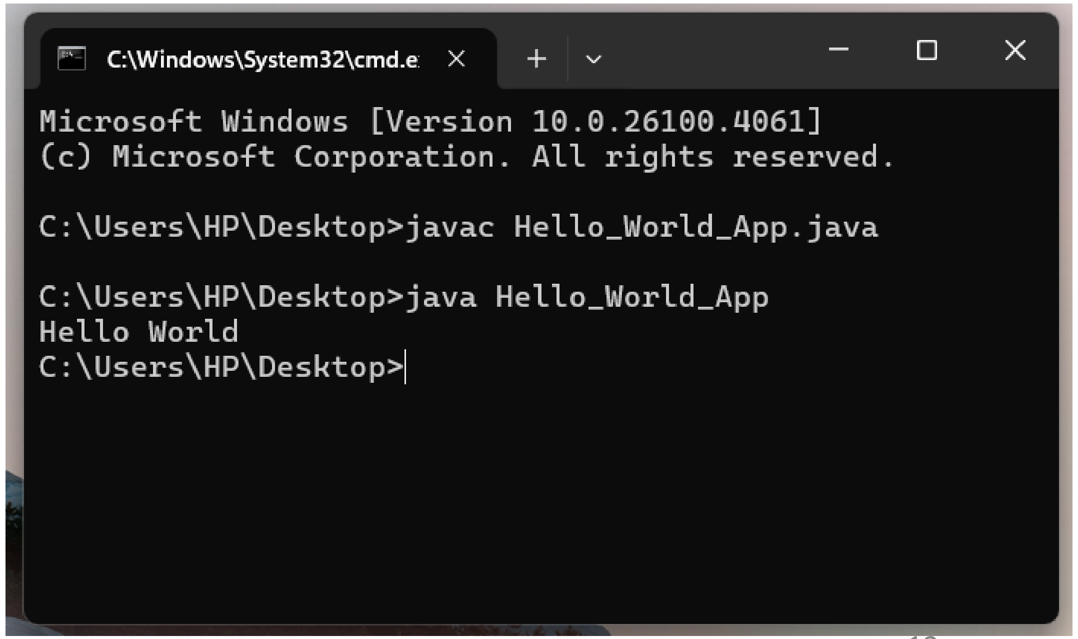
2. Compile your code as follows:

   javac HelloWorldApp.java

Both the compiler (javac) and launcher tool (java) are *case-sensitive*,

3. To run your program:

   java HelloWorldApp



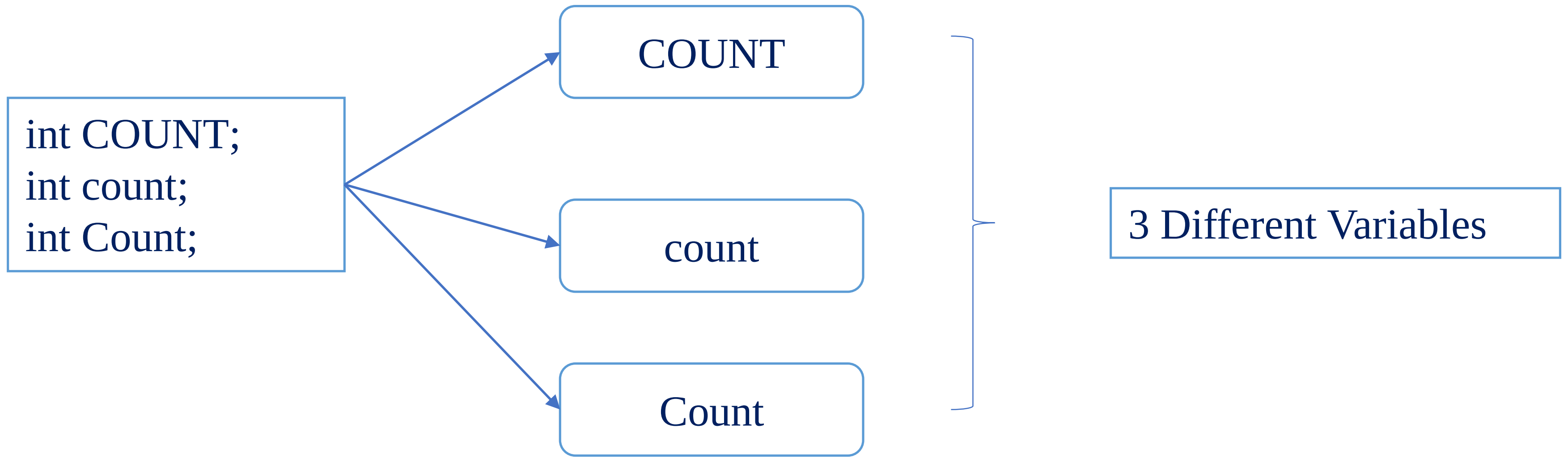
```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\Desktop>javac Hello_World_App.java

C:\Users\HP\Desktop>java Hello_World_App
Hello World
C:\Users\HP\Desktop>
```

As a Good Software Engineer…

Component	**Rule / Convention**	**Example**
Package Declaration	All lowercase; starts the file	package com.example.myapp;
Class Declaration	Use PascalCase (CapitalCase)	public class HelloWorld { }
Method Declaration	Use camelCase; should describe the action	public static void main(String[] args) { }
Single-line Comment	Start with //	// This prints Hello World
Multi-line Comment	Start with /* and end with */	/* This is a multi-line comment */
Documentation Comment	Start with /** and use @ tags (for Javadoc)	/** This is a doc comment \n * @author Nimesha */
Print Statement	Java statement to display output	System.out.println("Hello World!");
File Name Rule	Must match the public class name	Class: HelloWorld File: HelloWorld.java

Java is Case-Sensitive

- This means that a variable named as Count, for example, would NOT be the same as a variable named as count or COUNT.
- So, if you get an error message telling you that an identifier has not been declared, and you think you have declared it, check the case!

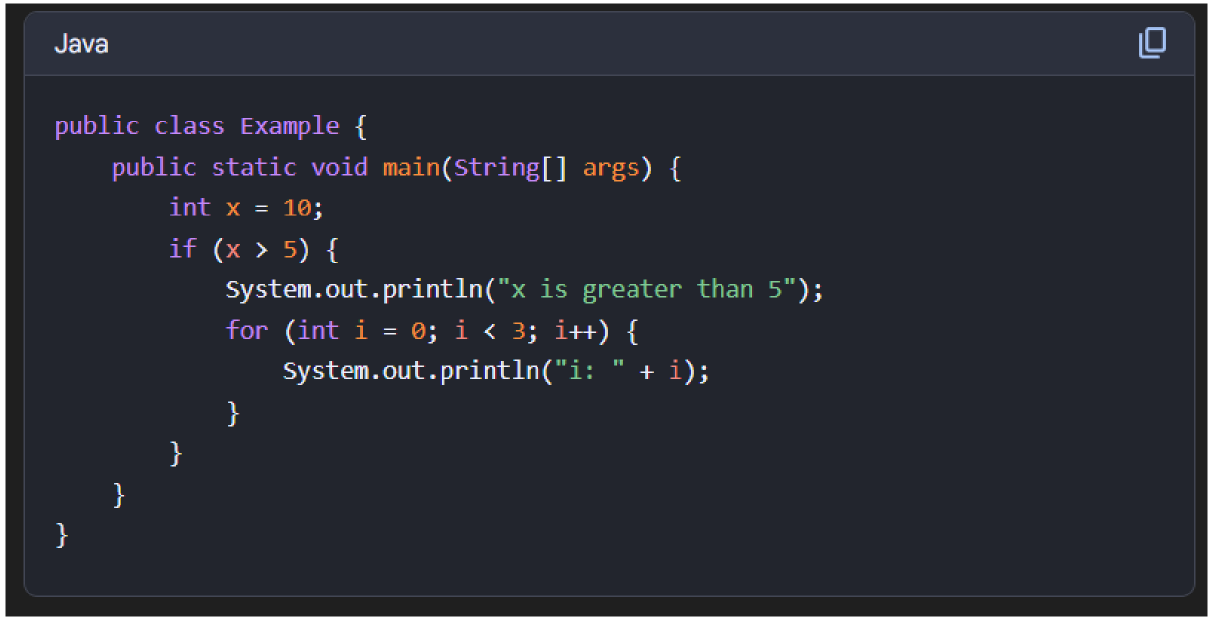


Coding style

- Statements are **terminated** by **“;”** characters.
- **Blocks** start with **{** and end with **}**
- A statement can be split over several lines – you can help make the code clearer and avoid untidy word wrap!
- **Blank lines** between statements improve clarity.
- Use **indenting** within structures for clarity.

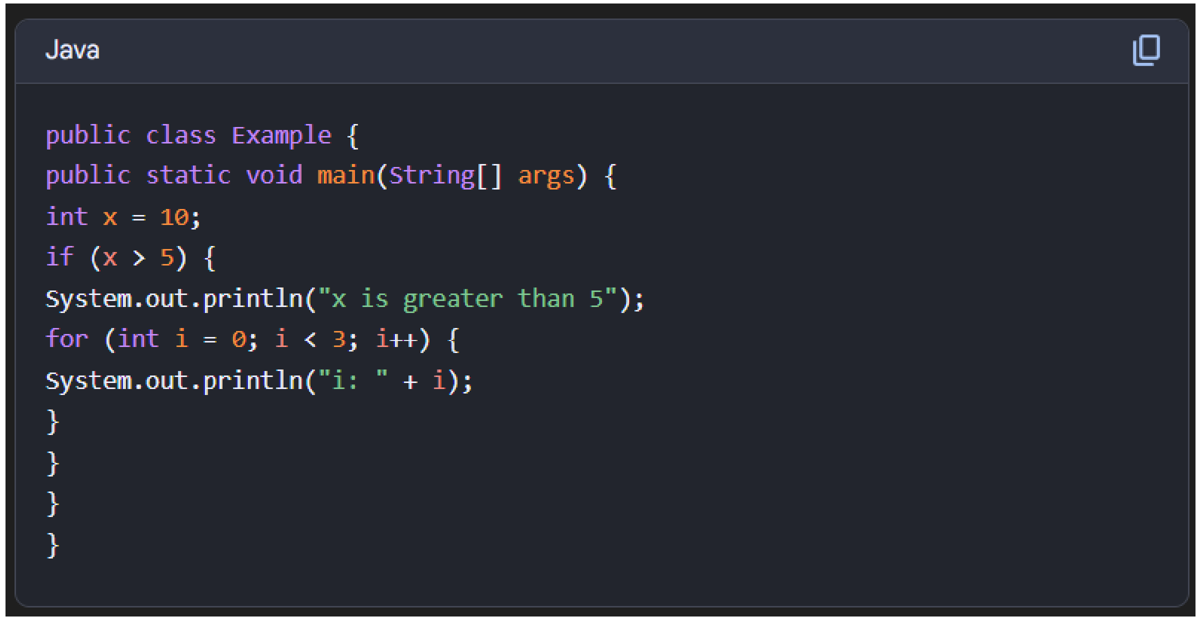
Best Practices

Which one you choose



Java

```java
public class Example {
    public static void main(String[] args) {
        int x = 10;
        if (x > 5) {
            System.out.println("x is greater than 5");
            for (int i = 0; i < 3; i++) {
                System.out.println("i: " + i);
            }
        }
    }
}
```



Java

```java
public class Example {
public static void main(String[] args) {
int x = 10;
if (x > 5) {
System.out.println("x is greater than 5");
for (int i = 0; i < 3; i++) {
System.out.println("i: " + i);
}
}
}
}
```

Thank you!