

# **Advanced Programming**

## **Java Introduction**

**Lecturer:** Ali Najimi

**Author:** Hossein Masihi

**Faculty of Computer Engineering**

**Sharif University of Technology**

**Fall 2025**





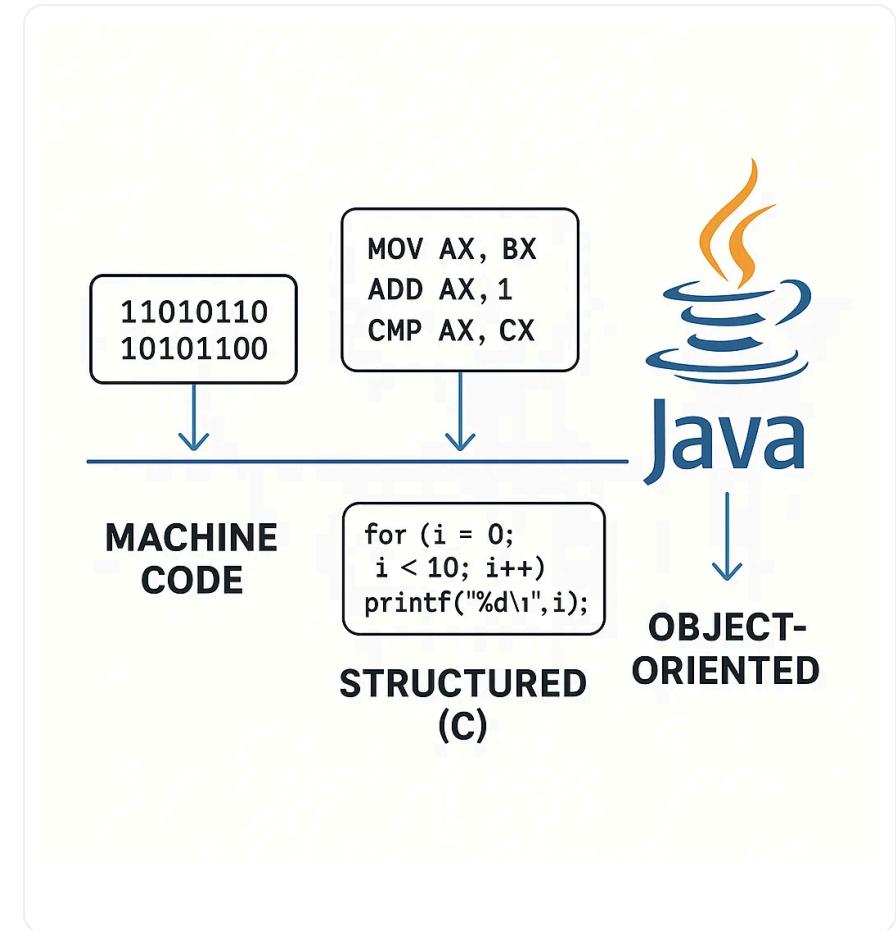
# Table of Contents

1. Programming History
2. What is Java?
3. JVM (Java Virtual Machine)
4. Java Versions
5. Platforms
6. Executables & Extensions
7. Popularity & Usage
8. Frameworks & Tooling
9. Why Java?
10. Hello World!



# Programming History

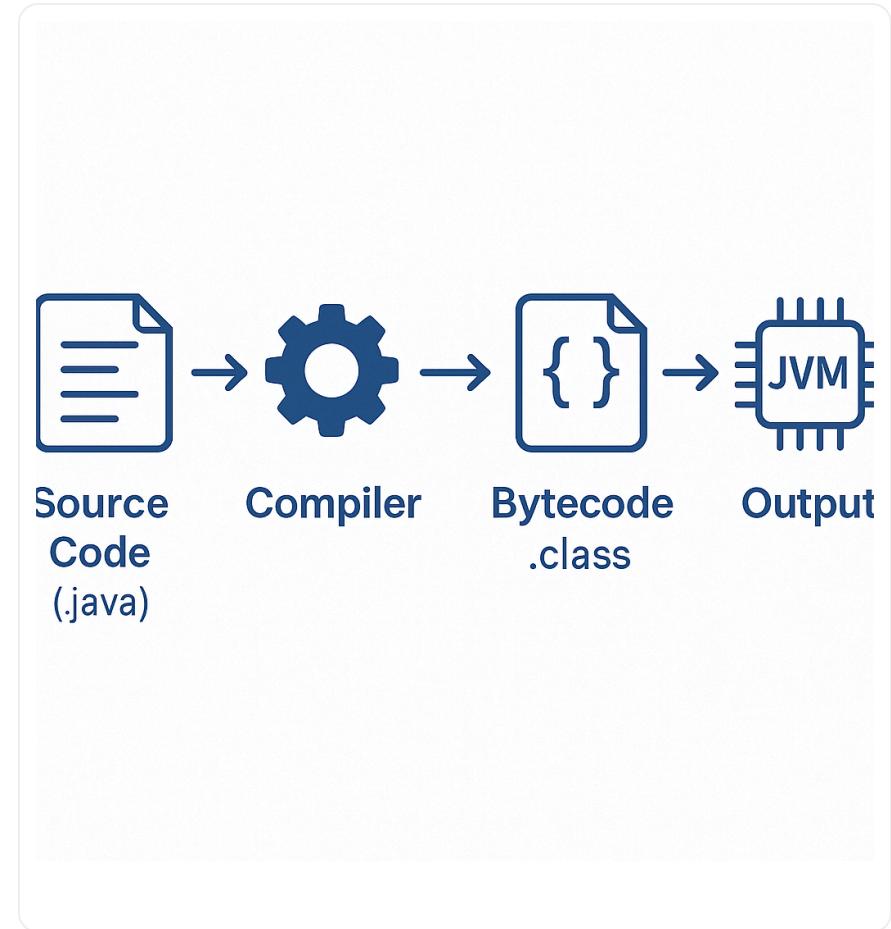
- Early languages: **Machine** → **Assembly** → **Structured** → **OOP**
- Machine & Assembly were hardware-dependent.
- Structured Programming (C) introduced functions & flow control.
- Object-Oriented Programming (Java, C++) introduced **classes** & **objects**.
- Java emerged in **1995** by *James Gosling* (Sun Microsystems).





# What is Java?

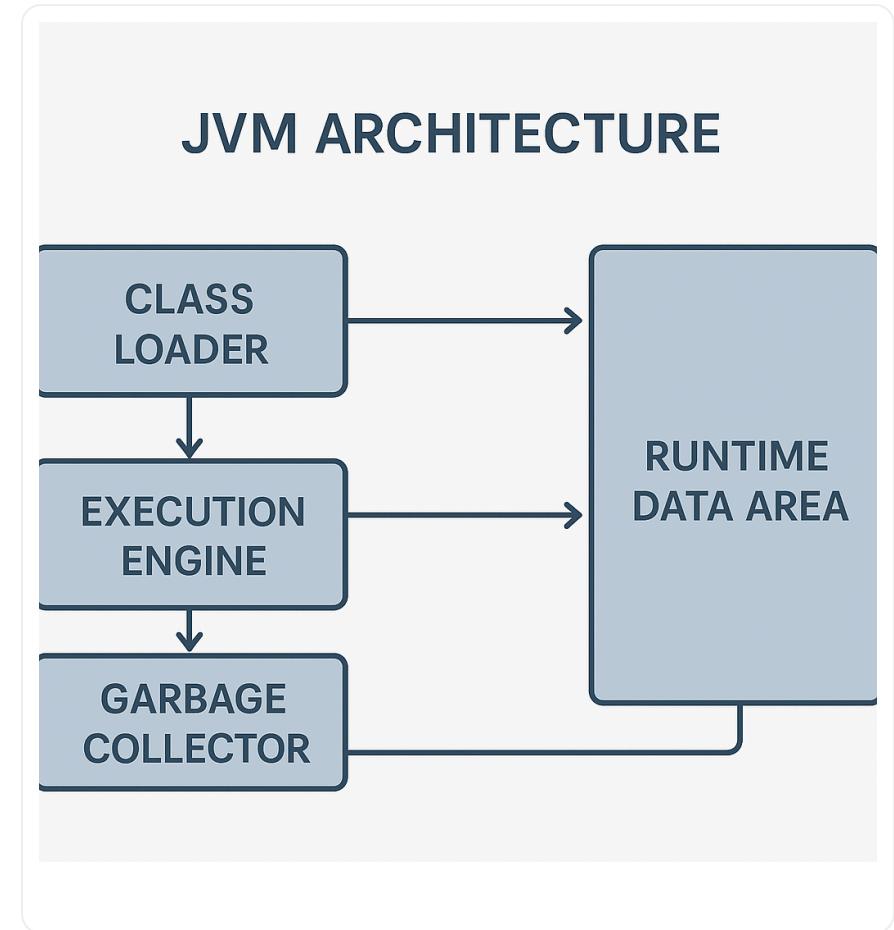
- High-level, Object-Oriented, Platform-Independent language.
- Source code compiled to **Bytecode**, executed by JVM.
- Motto: “**Write Once, Run Anywhere.**”
- Designed for portability, reliability, and security.





# JVM (Java Virtual Machine)

- JVM converts bytecode → machine instructions.
- Provides abstraction from underlying hardware.
- Components:
  - Class Loader
  - Execution Engine
  - Garbage Collector
- Enables the same code to run across Windows, Linux, and macOS.





## Java Versions

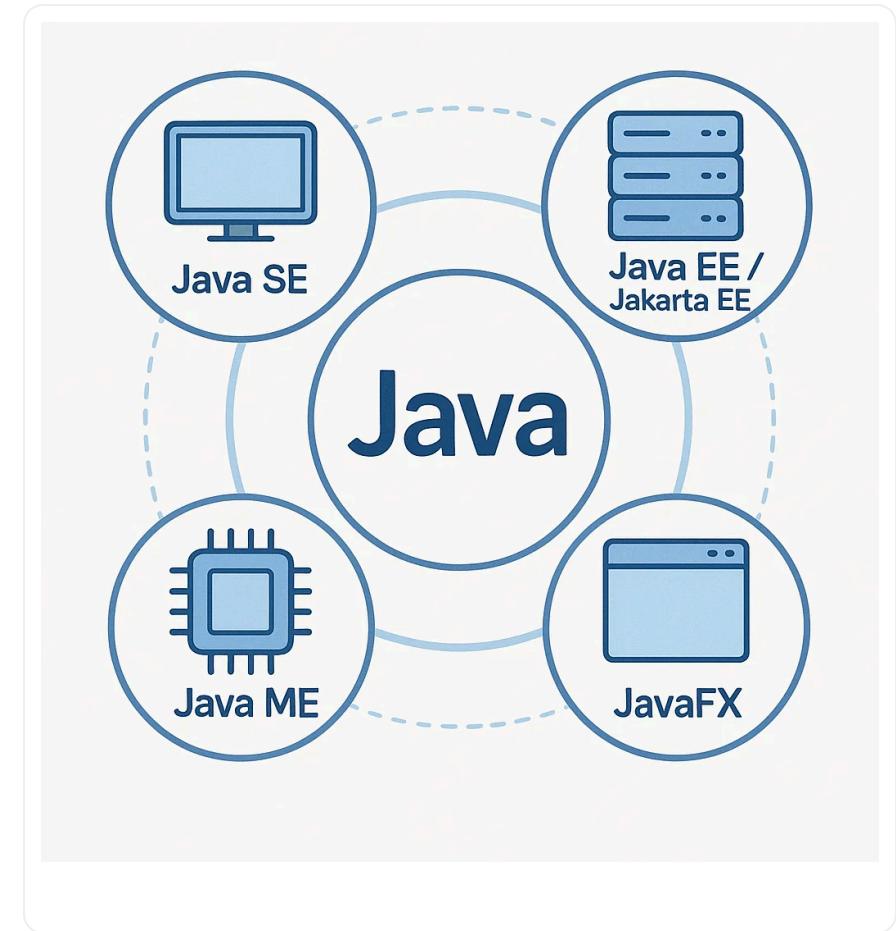
- **Java 1.0 (1995)** – initial release
- **Java 5 (2004)** – generics & annotations
- **Java 8 (2014)** – LTS, introduced lambdas
- **Java 11 (2018)** – LTS, modern baseline
- **Java 17 (2021)** – LTS, sealed classes & records
- **Java 21 (2023)** – LTS, virtual threads
- **Java 25 (2025)** – LTS, pattern matching enhancements





# Platforms

- **Java SE:** Standard Edition — desktop apps.
- **Jakarta EE (formerly Java EE):** Enterprise — web & servers.
- **Java ME:** Micro Edition — embedded & mobile.
- **JavaFX:** GUI toolkit for desktop applications.

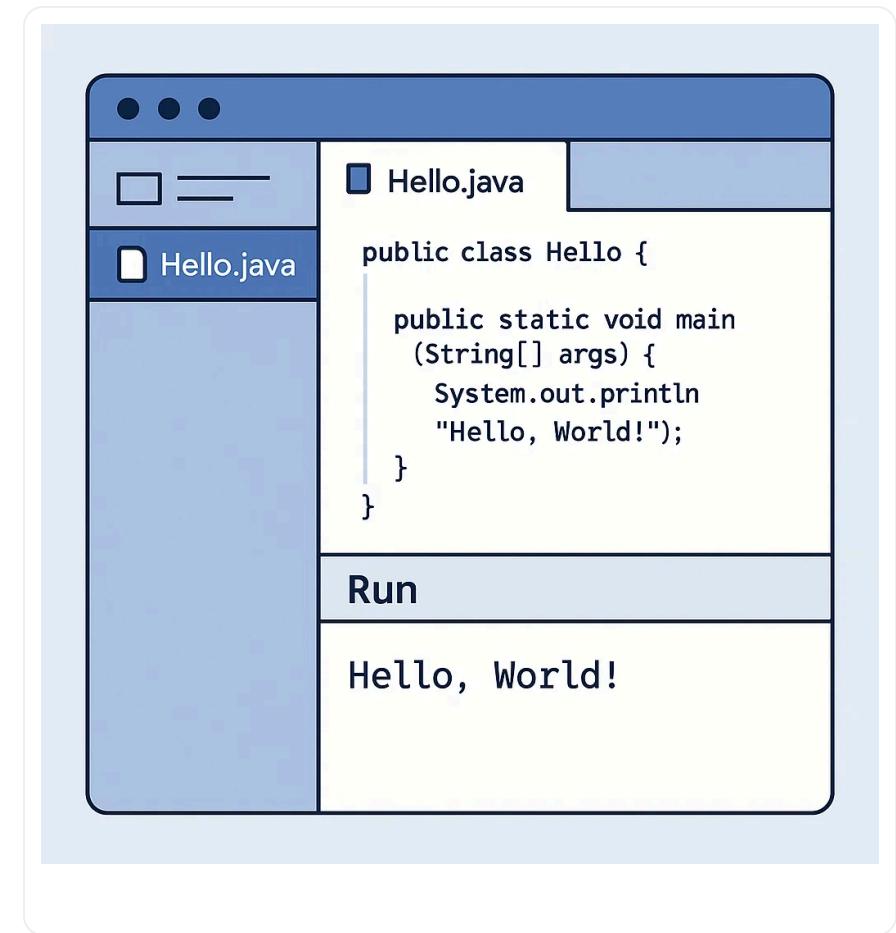




# Executables & Extensions

- Common Java file types:
  - `.java` → source code
  - `.class` → compiled bytecode
  - `.jar` → Java Archive
  - `.war` → Web Archive
  - `.jmod` → Java module
- Compile & run:

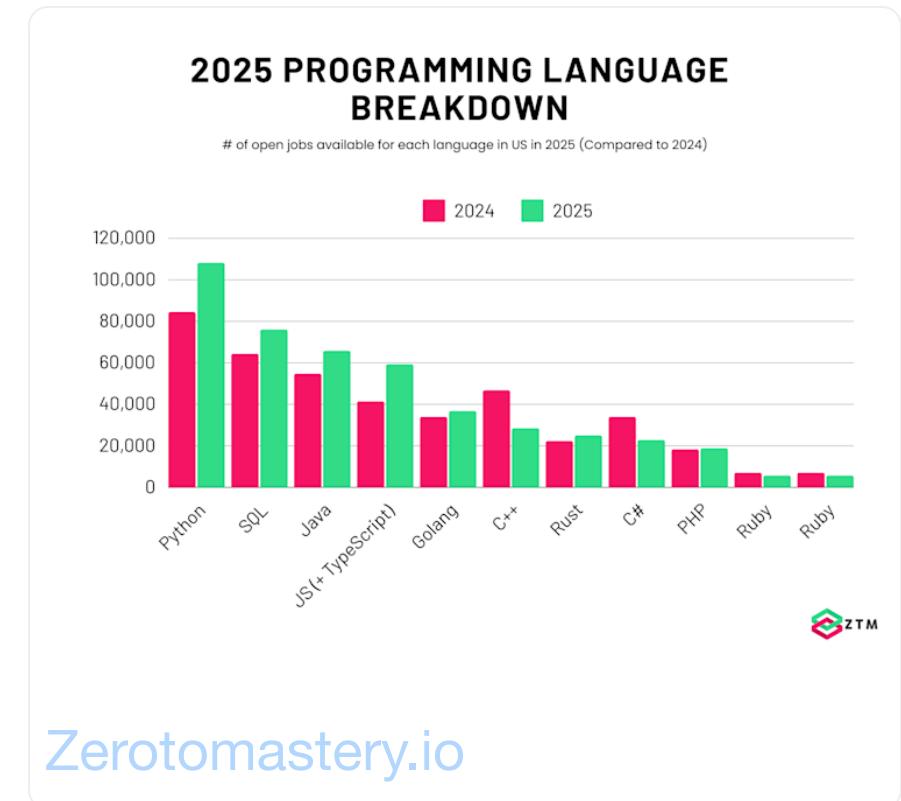
```
javac Hello.java  
java Hello
```

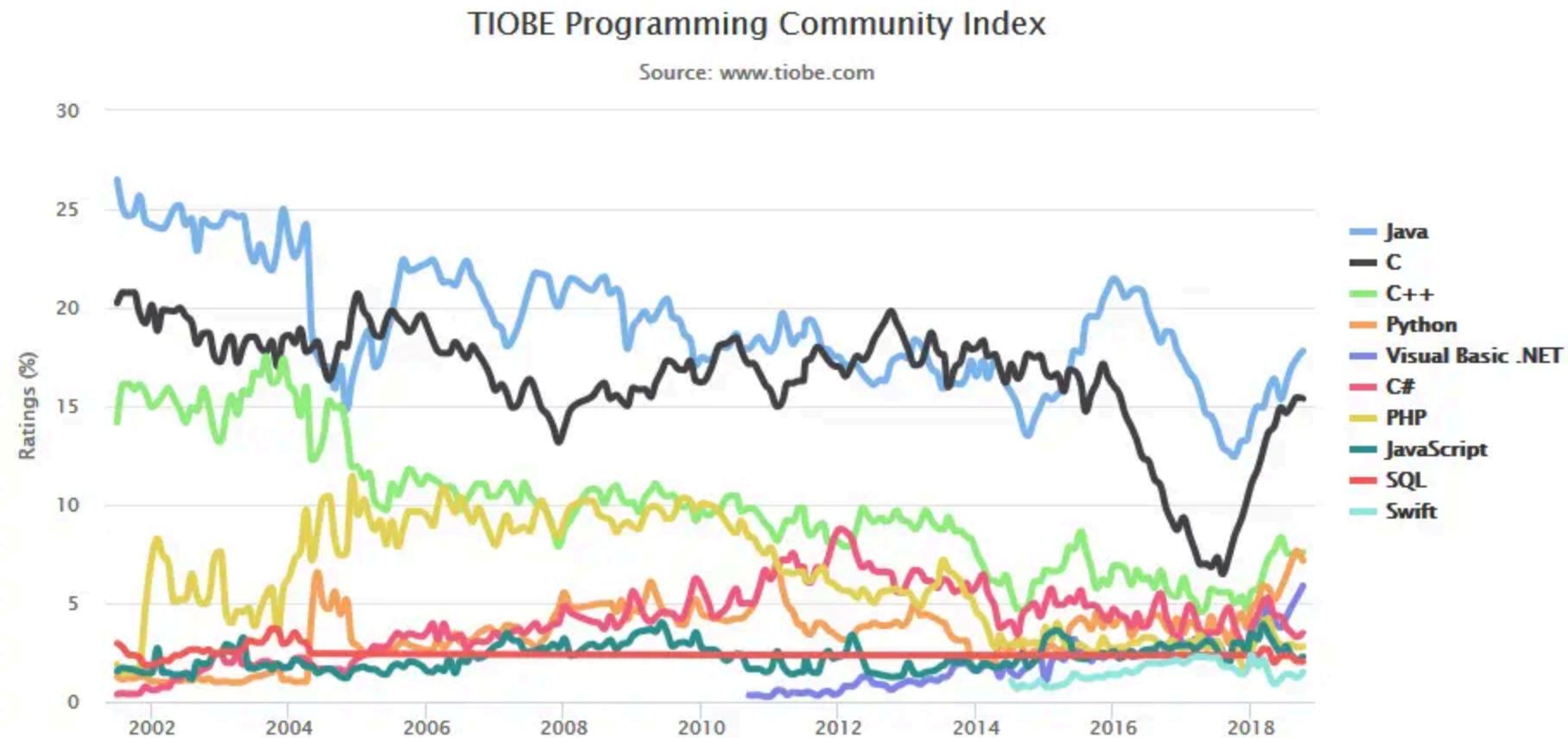




## Popularity & Usage

- Java remains among the top languages globally.
- Heavy use in enterprise, Android (legacy), and cloud services.
- Ranking factors include community size, job demand, and ecosystem.
- Stable presence alongside Python and JavaScript.





i-programmer.info



# Frameworks & Tooling

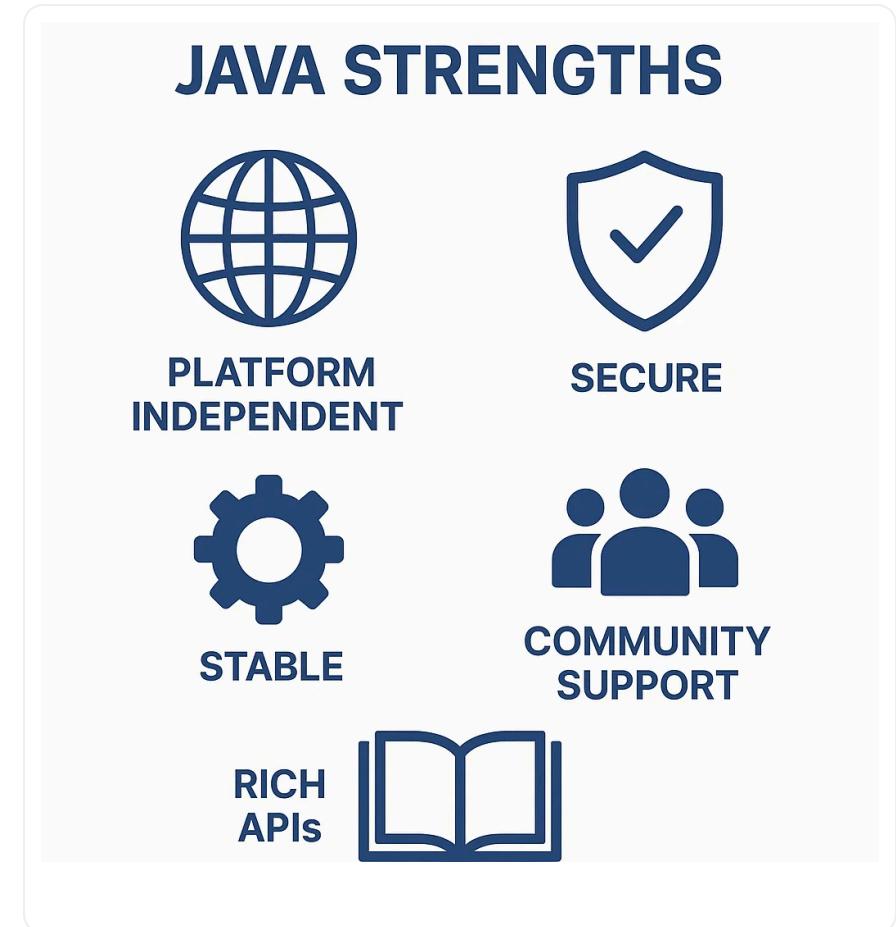
- **Spring Boot** — web & enterprise.
- **Hibernate** — ORM.
- **Maven / Gradle** — build automation.
- **JUnit** — testing framework.





# Why Java?

- Platform independence
- Strong memory management
- Large community & ecosystem
- Secure and stable
- Rich standard libraries
- Backward compatibility

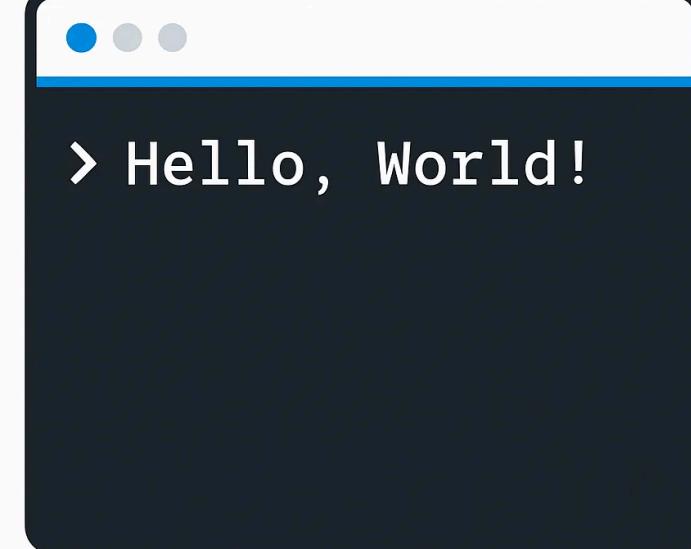




# Hello World!

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

- Entry point: `main(String[] args)`
- `System.out.println` writes to stdout
- Class name matches file name





**Thank You** 🙌

AP – Java Introduction

*Advanced Programming – Fall 2025 – Sharif University of Technology*