

# Java Overview, History & Features

## Overview

Welcome to the first step in your journey to mastering Java! Before we write complex algorithms or build full-scale applications, we need to understand the tools we are working with. In this lesson, we cover what Java is, its fascinating history, its core superpowers, and a first look at the mandatory code skeleton required to run any Java program.

## 1. What is Java & The "WORA" Concept

Java is a high-level, class-based, object-oriented programming language. It is the powerhouse behind millions of Android applications, enterprise-level backend servers, web applications, and even video games.

The biggest advantage of Java is a concept called **WORA**: *Write Once, Run Anywhere*.

Unlike older languages that require you to rewrite code for different operating systems, Java code is compiled into an intermediate format called "Bytecode." This means you can write your Java code once on a Windows machine, and it will run perfectly on Mac or Linux without changing a single line.

## 2. A Brief History of Java

Understanding where a language came from helps you understand why it is built the way it is.

- **The Beginning:** Java was initiated in 1991 by James Gosling and his "Green Team" at Sun Microsystems.
- **The Name:** It was originally called "Greentalk", and later "Oak" (named after an oak tree outside Gosling's office). Because "Oak" was already trademarked, the team renamed it to "Java", inspired by the Indonesian coffee they drank while working. This is why the Java logo is a steaming cup of coffee!
- **Today:** Sun Microsystems was acquired by Oracle Corporation, meaning Oracle currently owns and maintains Java.

### 3. Key Features of Java

Why do massive tech companies still rely heavily on Java today? Because of these four core features:

- **Simple:** Java removed the confusing and dangerous features of C++ (like explicit memory pointers), making it much easier to write and maintain clean code.
- **Object-Oriented:** Everything in Java is modeled as "Objects" and "Classes." This makes it incredibly easy to reuse code and manage massive software projects.
- **Robust & Secure:** Java handles memory management for you through "Automatic Garbage Collection." It also runs in a secure, virus-free virtual environment, preventing malicious software from attacking the host computer.
- **Multithreaded:** Java can run multiple tasks simultaneously in the background (threads), allowing developers to build highly efficient and fast applications that maximize CPU performance.

### 4. The Java Skeleton (Your First Look)

Just like HTML requires a boilerplate (<html>, <head>, <body>), every single Java program requires a strict foundational skeleton to run.

Here is the absolute minimum code required to print a message to the screen:

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

- **public class Main:** In Java, all code must live inside a "Class." Think of this as the main container for your file.
- **public static void main(String[] args):** This is the most important line for a beginner. This is the **Main Method**. It acts as the front gate to your program. When you tell the computer to run your code, it immediately looks for this exact line to start executing.