Core Java: The Introduction

1. What is Java?

- Java is a **high-level, object-oriented, class-based programming language** developed by **James Gosling** at **Sun Microsystems** in **1995**.
- It is **platform-independent**, **secure**, **portable**, and designed for building robust, distributed, and cross-platform applications.

2. Java Architecture Overview

◆ Java Code Execution Process:

1. **Write Code:** .java file using editor.
2. **Compile:** Using javac, converts to **Bytecode** (.class file).
3. **Execute:** Bytecode is run by the **Java Virtual Machine (JVM)**.

◆ Key Components:

Component	Description
JDK (Java Development Kit)	Complete package for development (includes JRE, JVM, tools like compiler javac, debugger, etc.).
JRE (Java Runtime Environment)	Provides environment to run Java programs (includes JVM + libraries).
JVM (Java Virtual Machine)	Executes bytecode, ensures portability. Converts bytecode to machine code at runtime.
Garbage Collector	Automatic memory management tool in JVM that reclaims memory from unused objects.

3. Features of Java (Java Buzzwords)

Feature	Description
Simple	Easy to learn, clean syntax, no pointers.
Object-Oriented	Everything is an object (except primitives).
Platform Independent	Write Once, Run Anywhere (WORA) using bytecode and JVM.
Secure	Bytecode verification and runtime security checks.
Robust	Strong memory management, exception handling.
Multithreaded	Built-in support for concurrent execution.
Distributed	Supports networked applications via RMI and Socket programming.
High Performance	Faster with **Just-In-Time (JIT)** compiler.
Dynamic	Loads classes at runtime, can interact with native libraries.

4. Java vs Other Programming Languages

Criteria	Java	C++	Python
Platform Independence	Yes	No	Partial
Memory Management	Automatic (GC)	Manual	Automatic
Speed	Moderate	Fast	Slower
Verbosity	Verbose	Verbose	Concise
Performance	High (JIT)	Very High	Moderate
Compilation	Bytecode + JVM	Native Code	Interpreted/Bytecode
Use Cases	Web, Enterprise, Android, Backend	Systems, Games	AI, Data Science, Automation

---

5. Real-World Projects Built in Java

Company / Platform	Application
Google	Android SDK (built on Java)
Netflix	Backend microservices
Amazon	Web and enterprise systems
Twitter	Core services

Company / Platform	Application
NASA WorldWind	3D virtual globe (Java-based SDK)
Minecraft	Popular game written in Java
Hadoop	Big Data framework written in Java

6. Java-Based Programming Languages

These languages use Java as a foundation or compile to JVM bytecode:

Language	Description
Kotlin	Modern Android and backend programming, interoperable with Java.
Scala	Functional + object-oriented, runs on JVM.
Groovy	Dynamic language for scripting and web apps.
Clojure	Functional Lisp dialect on the JVM.
Jython	Python implemented on JVM.

7. Interesting Java Facts

- Over **3 billion devices** run Java.
- Java is the **#1 language** for enterprise applications.
- Used extensively in **banking, telecom, insurance, e-commerce**.
- **Oracle** owns Java now (acquired Sun Microsystems in 2010).
- Java has been **consistently in top 3** languages in TIOBE and GitHub rankings.
- Java powers **smartcards, ATMs, POS terminals, set-top boxes**.

8. Summary for First Lecture

- Java is a robust, secure, and platform-independent programming language.
- JVM makes Java portable across systems.
- JDK → JRE → JVM are core for development and execution.
- Garbage collection improves memory safety.
- Java's features make it versatile across many industries.
- Widely adopted with strong community and ecosystem.