👉🏻

Introduction to Java

▼ Why you must learn Java?

1. **Versatility:** Java is versatile and used in various domains, including web development, mobile app development (Android), enterprise systems, and more.
2. **Platform Independence:** Java's "write once, run anywhere" principle allows code to run on different platforms without modification.
3. **Community and Resources:** Java has a vast and supportive community, offering abundant resources, libraries, and frameworks for developers.
4. **Object-Oriented Programming:** Java's strong emphasis on object-oriented programming provides a solid foundation for understanding key programming concepts.
5. **Job Opportunities:** Java is widely used in the industry, leading to numerous job opportunities and a strong demand for Java developers.
6. **Scalability and Performance:** Java's scalability and performance make it suitable for building robust and high-performance applications.

▼ What is a Programming Language?

- Humans use natural language (like Hindi/English) to communicate.
- Computers only understand 0/1 or on/off.
- Giving instructions to a computer

    - Instructions: Tells computer what to do.
    - These instructions are called code.
    - Human instructions are given in High level languages.

Compiler converts high level languages to low level languages or machine code.

▼ What is algorithm?

An algorithm is a step-by-step procedure or set of rules designed to perform a specific task or solve a particular problem.

▼ What is Syntax?

Syntax refers to the set of rules that dictate the combination of symbols and words to create valid and meaningful statements in a programming language. It defines the structure and format that code must follow for the computer to understand and execute it correctly. In essence, syntax governs the proper arrangement of elements in a programming language to ensure the code is both accurate and interpretable by the computer.

▼ What is Java?

Java is a popular programming language, **created in 1995**.

It is owned by **Oracle**, and more than **3 billion** devices run Java.

It is used for:

- Mobile applications (specially Android app, Desktop applications, Web applications, Web servers and application servers, Games, Database connection, And much, much more!

▼ History of Java?

Java, developed by **James Gosling** and his team at **Sun Microsystems**, has a rich history:

1. **Origins (1991-1995):** Java's development began in 1991 as a project called "Oak" by James Gosling, Mike Sheridan, and Patrick Naughton. Its initial focus was on creating software for consumer electronic devices. In 1995, Oak was renamed Java, and the first public version of Java, JDK 1.0, was released.

2. **Platform Independence (1995):** One of Java's groundbreaking features was its **"Write Once, Run Anywhere" (WORA)** capability, allowing Java programs to run on any device with a Java Virtual Machine (JVM). This **independence from hardware and operating systems** contributed to its popularity.

3. **Java 2 (1998):** The introduction of Java 2 (J2SE 1.2) brought significant enhancements, including the Swing GUI toolkit, the Collections Framework, and the "assert" keyword.
4. **Enterprise Java (EJB, J2EE):** Java's expansion into enterprise computing came with the Enterprise JavaBeans (EJB) and the Java 2 Enterprise Edition (J2EE) in the late 1990s. This facilitated the development of scalable and distributed enterprise applications.
5. **Open Sourcing (2006):** Sun Microsystems open-sourced Java under the GNU General Public License (GPL) in 2006, making it more accessible to developers and enabling community-driven development.
6. **Oracle Acquisition (2010):** Oracle Corporation acquired Sun Microsystems in 2010, becoming the steward of Java. Despite concerns, Java's development continued, and subsequent versions were released.
7. **Java 8 and Beyond:** Java 8 (2014) introduced lambdas, the Stream API, and other features, marking a significant shift in the language. Subsequent releases, such as Java 9, 10, and later, focused on modularity, performance improvements, and new language features.
8. **Project Loom and Valhalla:** Ongoing projects like Loom (aimed at improving concurrency with lightweight threads) and Valhalla (focused on value types and performance enhancements) demonstrate Java's commitment to staying relevant and evolving.

▼ Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

▼ Java Buzzwords

Java is associated with several key buzzwords that highlight its features and characteristics. Some of the prominent Java buzzwords include:

1. **Robust:** Java incorporates features such as strong type-checking, exception handling, and automatic memory management (garbage collection) to enhance program robustness.
2. **Multithreaded:** Java facilitates concurrent programming by supporting multithreading, enabling the execution of multiple threads of a program simultaneously.
3. **Architectural-Neutral:** Java is designed to be architecture-neutral, meaning that the compiler generates bytecode that can run on any Java-enabled device.
4. **High Performance:** Although Java is often associated with platform independence, it also focuses on delivering high performance through just-in-time (JIT) compilation and other optimization techniques.
5. **Distributed:** Java supports the development of distributed computing applications, allowing components to be distributed across multiple networked computers.
6. **Object-Oriented:** Java follows the principles of object-oriented programming (OOP), emphasizing the organization of code into reusable and modular structures called objects.
7. **Simple:** Java is designed to be easy to learn and use. Its syntax is clean and straightforward, reducing complexity for developers.
8. **Platform-Independent:** Java programs can run on any device with a Java Virtual Machine (JVM), providing platform independence and enabling the "Write Once, Run Anywhere" (WORA) concept.
9. **Secure:** Java's design includes security features to create a secure computing environment. This includes the use of the Java Security Manager and other mechanisms to protect against malicious code.
10. **Dynamic:** Java supports dynamic loading of classes and dynamic memory allocation, contributing to greater adaptability and flexibility in program execution.