Java is a high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle) in 1995. It follows the principle of **"Write Once, Run Anywhere" (WORA)** due to its platform-independent nature, achieved through the **Java Virtual Machine (JVM)**. Java is widely used in enterprise applications, web development, mobile development (Android), and cloud-based applications.

Key Concepts in Java

1. **Object-Oriented Programming (OOP)** – Java follows OOP principles such as:
   - **Encapsulation**: Wrapping data and methods within a class to protect it from outside access.
   - **Abstraction**: Hiding implementation details and exposing only the necessary parts.
   - **Inheritance**: Allowing one class to acquire the properties of another.
   - **Polymorphism**: Enabling a single function to behave differently based on the context.
2. **Memory Management & Garbage Collection** – Java has an automatic garbage collector that removes unused objects from memory, improving performance and preventing memory leaks.
3. **Multithreading** – Java provides built-in support for multithreading, allowing parallel execution of tasks using the **Thread** class and **Runnable** interface.
4. **Exception Handling** – Java includes a robust error-handling mechanism with **try, catch, finally**, and **throw/throws** keywords to manage runtime errors effectively.
5. **Java API & Libraries** – Java offers a vast standard library covering collections, networking, file handling, concurrency, and more. The **Java Standard Edition (SE)** provides core functionalities, while **Java Enterprise Edition (EE)** extends it for web and enterprise applications.
6. **JVM, JRE, and JDK** – Java is executed using:
   - **JVM (Java Virtual Machine)**: Converts bytecode into machine code for execution.
   - **JRE (Java Runtime Environment)**: Provides libraries and the JVM necessary for running Java applications.
   - **JDK (Java Development Kit)**: Includes the JRE plus development tools like the compiler (javac).
7. **Frameworks & Ecosystem** – Java has a strong ecosystem with frameworks like **Spring Boot** for backend applications, **Hibernate** for database interactions, and **JavaFX** for GUI applications.

8. **Security & Performance –** Java includes built-in security features like the **Security Manager**, **Cryptography API**, and **Access Control Lists (ACLs)**, making it suitable for banking and enterprise applications.

9. **Applications of Java –** Java is widely used in:

    - **Enterprise applications** (Spring Boot, Microservices)
    - **Mobile development** (Android development with Java/Kotlin)
    - **Web applications** (Java Servlets, JSP, Spring MVC)
    - **Cloud computing** (AWS, Google Cloud using Java-based services)
    - **Big data & machine learning** (Hadoop, Apache Spark, Weka)