Prerequisites

1. Core Java

- Fundamental features like **OOP concepts**, **Java APIs**, **basic I/O**, and **memory management**.
- Key topics: classes, objects, inheritance, polymorphism, encapsulation, abstraction.

2. Exception

- Mechanism to handle runtime errors using try, catch, finally, throw, and throws.
- Types: Checked and Unchecked exceptions.

3. Threads

- Enables multitasking; created using Thread class or Runnable interface.
- Synchronization for thread safety in shared resources.

4. Collections

- Data structures like **List**, **Set**, **Map**.
- Common classes: ArrayList, HashSet, HashMap.

5. JDBC

- Java API to interact with databases.
- Supports SQL operations: SELECT, INSERT, UPDATE, DELETE via Connection, Statement, PreparedStatement.

6. Maven/Gradle

- Build tools for managing dependencies, project lifecycle, and automation.
- **Maven** uses pom.xml, **Gradle** uses build.gradle for configurations.

7. Spring ORM/Hibernate

- **Spring ORM** integrates Hibernate to manage database operations.
- **Hibernate** is an ORM framework that maps Java objects to database tables.

8. Servlet

- Java class that handles HTTP requests and responses in web applications.
- Part of the **Java EE** framework; works with HttpServletRequest and HttpServletResponse.