## 🧱 1. Maven Fundamentals

- What is Maven and why it's used
- Maven vs Gradle (basic comparison)
- Maven lifecycle phases: validate, compile, test, package, verify, install, deploy
- Maven directory structure (src/main/java, src/test/java, etc.)

---

## 📦 2. pom.xml Mastery

- Declaring dependencies
- Dependency scopes: compile, provided, runtime, test, system, import
- Managing transitive dependencies
- Using dependency management (<dependencyManagement>)
- Parent POMs and inheritance

---

## 🔧 3. Maven Plugins

- Common plugins:
  - maven-compiler-plugin
  - spring-boot-maven-plugin
  - maven-surefire-plugin (for testing)
  - maven-jar-plugin, maven-war-plugin
- Plugin configuration and execution goals

---

## 🧩 4. Profiles and Environments

- Creating custom build profiles (<profiles>)
- Switching between environments (dev, test, prod)
- Externalizing configuration using properties files

---

## 🧪 5. Testing with Maven

- Running unit and integration tests
- Configuring test reports
- Skipping tests (-DskipTests, -Dmaven.test.skip=true)

---

### 🧭 6. Multi-Module Projects

- Structuring large applications with multiple modules
- Parent-child module relationships
- Aggregator vs inheritance

---

### 🚀 7. Maven in CI/CD

- Using Maven in Jenkins, GitHub Actions, GitLab CI
- Automating builds, tests, and deployments
- Artifact versioning and publishing to repositories

---

### 🌐 8. Maven Repositories

- Local, central, and remote repositories
- Nexus or Artifactory for private repo management
- Snapshot vs release versions

---

### 🧠 9. Advanced Concepts

- BOM (Bill of Materials) for dependency version control
- Maven wrapper (mvnw) for consistent builds
- Effective POM (mvn help:effective-pom)
- Maven goals vs phases