Exp-1

Experiment 1: Introduction to Maven and Gradle

Introduction to Maven and Gradle: Overview of Build Automation Tools, Key Differences Between Maven and Gradle, Installation and Setup

Objective

To understand build automation tools, compare **Maven** and **Gradle**, and set up both tools for software development.

1. Overview of Build Automation Tools

Build automation tools simplify the process of **compiling, testing, packaging, and deploying** software projects. They manage dependencies, execute tasks, and integrate seamlessly with CI/CD pipelines.

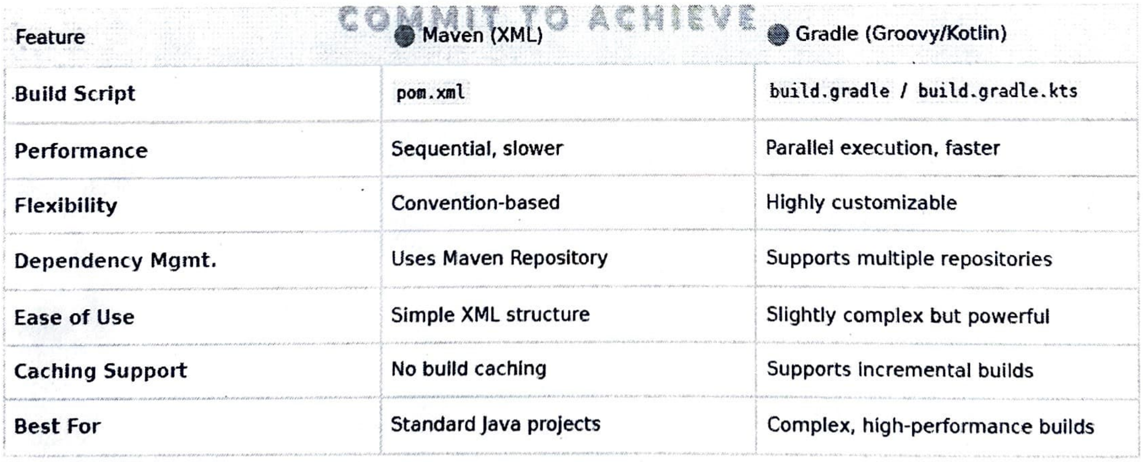
Why Use Build Automation?

☑ Ensures **consistency** across builds
☑ Handles **dependency management** automatically
☑ Reduces **manual errors** and increases efficiency
☑ Integrates with tools like **Jenkins & Azure DevOps**

Popular Build Automation Tools

Apache Ant → Script-based, manual dependency management
Apache Maven → XML-based, convention-over-configuration model
Gradle → Flexible, fast, and supports Groovy/Kotlin DSL

2. Key Differences Between Maven and Gradle



Feature	● Maven (XML)	● Gradle (Groovy/Kotlin)
Build Script	`pom.xml`	`build.gradle / build.gradle.kts`
Performance	Sequential, slower	Parallel execution, faster
Flexibility	Convention-based	Highly customizable
Dependency Mgmt.	Uses Maven Repository	Supports multiple repositories
Ease of Use	Simple XML structure	Slightly complex but powerful
Caching Support	No build caching	Supports incremental builds
Best For	Standard Java projects	Complex, high-performance builds

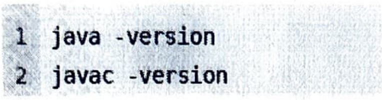
Maven is great for structured, **enterprise-level** projects.
Gradle is ideal for **scalable and performance-driven** applications.

3. Installation and Setup

● 3.1 Installing Maven

☑ **Step 1: Install Java (JDK 17 Recommended)**

Check if Java is installed:



```
java -version
javac -version
```

☑ **Step 2: Download and Install Maven**

Download from: Download Apache Maven – Maven
Extract it to a folder (e.g., C:\Maven).

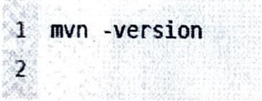
☑ **Step 3: Configure Environment Variables**

Windows:

Add MAVEN_HOME → C:\Maven\apache-maven-<version>
Add M2_HOME → C:\Maven\apache-maven-<version>
Update Path → %MAVEN_HOME%\bin

☑ **Step 4: Verify Installation**

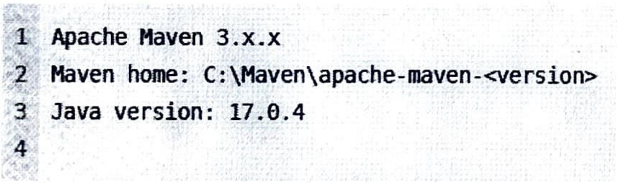
Run:



```
mvn -version

```

Expected Output:



```
Apache Maven 3.x.x
Maven home: C:\Maven\apache-maven-<version>
Java version: 17.0.4

```

3.2 Installing Gradle

☑ **Step 1: Install Java (JDK 17 Recommended)**

Same steps as Maven.

☑ **Step 2: Download and Install Gradle**

Download from: Gradle | Releases
Extract it to a folder (e.g., C:\Gradle).

☑ **Step 3: Configure Environment Variables**

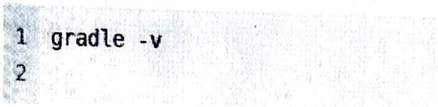
Windows:

Add GRADLE_HOME → C:\Gradle\gradle-<version>
Update Path → %GRADLE_HOME%\bin

Exp-1

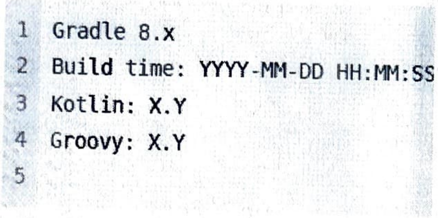
☑ Step 4: Verify Installation

Run:



```
gradle -v
```

Expected Output:



```
Gradle 8.x
Build time: YYYY-MM-DD HH:MM:SS
Kotlin: X.Y
Groovy: X.Y
```

Assessment Questions

1. What are the **key advantages** of using a build automation tool?
2. Mention **three** major differences between Maven and Gradle.
3. How does Gradle achieve **faster build times** compared to Maven?
4. What are the **necessary environment variables** for setting up Maven and Gradle?
5. How do you **verify** that Maven and Gradle are installed correctly?