Lab 4- Practical Exercise: Build and Run a Java Application with Maven, Migrate the Same Application to Gradle.

1: Step 1: Creating a Maven Project

Using Command Line:

- To create a basic Maven project using the command line, you can use the following command:

```
mvn archetype:generate -DgroupId=com.example -DartifactId=maven-example -
DarchetypeArtifactId=maven-archetype-quickstart  -DinteractiveMode=false
```

This command creates a new Java application project with a sample **build.gradle** file.

2: Step 2: Open The pom.xml File

You can manually navigate the project folder named call maven-example and open the file pom.xml and copy the below code and paste it then save it.

(Note: In case if you not getting project folder then type command in your cmd.

cd maven-example – is use to navigate the project folder.

notepad pom.xml – is use to open pom file in notepad.)

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">

<modelVersion>4.0.0</modelVersion><groupId>com.example</groupId>

<artifactId>maven-example</artifactId>

<version>1.0-SNAPSHOT</version><dependencies>
```

```
<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies><build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>
```

3: Step 3: Open Java Code (App.java) File

- Open a file App.java inside the src/main/java/com/example/ directory.
- After opening the App.java copy the below code and paste it in that file then save it.

```
package com.example;public class App {

public static void main(String[] args) {

System.out.println("Hello, Maven");

System.out.println("This is the simple realworld example....");

int a = 5;

int b = 10;

System.out.println("Sum of " + a + " and " + b + " is " + sum(a, b));

}

public static int sum(int x, int y) {

return x + y;

}

}
```

Note: before building the project make sure you are in the project folder if not navigate the project folder type command in your command prompt **cd maven-example**.

4: Run the Project

To build and run this project, follow these steps:

Open the terminal in the project directory and run the following command to build the project.

```
mvn clean install
```

Run the program with below command:

```
mvn exec:java -Dexec.mainClass="com.example.App"
```

5: Migrate the Maven Project to Gradle

1. **Initialize Gradle**: Navigate to the project directory (**gradle-example**) and run:

```
gradle init
```

- It will ask **Found a Maven build. Generate a Gradle build from this? (default: yes) [yes, no]**
  - Type **Yes**
- **Select build script DSL:**
  - 1: Kotlin
  - 2: Groovy
  - Enter selection (default: Kotlin) [1..2]
    - Type **2**
- **Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no]**
  - Type **No**

2. Navigate the project folder and open **build.gradle** file then add the below code and save it.

```
plugins {

   id 'java'

}



group = 'com.example'
```

```
version = '1.0-SNAPSHOT'


repositories {
    mavenCentral()
}


dependencies {
    testImplementation 'junit:junit:4.12'
}


task run(type: JavaExec) {
    main = 'com.example.App'
    classpath = sourceSets.main.runtimeClasspath
}
```

6: Run the Gradle Project

Build the Project: In the project directory (gradle-example), run the below command to build the project:

```
gradlew build
```

Run the Application: Once the build is successful, run the application using below command:

```
gradlew run
```

7: Verify the Migration

- **Compare the Output:** Make sure that both the **Maven** and **Gradle** builds produce the same output:
  - **Maven Output**:

```
Hello, Maven

This is the simple realworld example....

Sum of 5 and 10 is 15
```

- **Gradle Output**:

```
Hello, Maven

This is the simple realworld example....

Sum of 5 and 10 is 15
```