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

**Part A: Build and Run a Java Application with Maven**

**Generate the Maven Project**

**Open IntelliJ IDEA Terminal Press Alt + F12 to open the terminal.**

create a new Maven project with the group ID com.example and the artifact ID HelloMaven.

Explore the Maven Project Structure

HelloMaven/

├── pom.xml

└── src

├── main

│ └── java

│ └── com

│ └── example

│ └── App.java

└── test

└── java

└── com

└── example

└── AppTest.java

**pom.xml:** The Maven configuration file (POM) that defines your project’s coordinates, dependencies, and plugins.

**src/main/java:** Contains your application’s source code.

**src/test/java:** Contains unit tests.

**2. Compile and Package the Project Run the following commands to build the project:**

* **mvn compile**
* **mvn package**

**What it does:** This command compiles the source code, runs tests, and packages your application into a JAR file (located in the target directory).

Expected Output:You should see a “BUILD SUCCESS” message along with information on the created JAR file.

**3.Run the Maven Application**

Run the JAR File To run the generated JAR file, use Build the Maven Project

Execute the following command:

**java -jar target\HelloMaven-1.0-SNAPSHOT.jar com.example.App**

What it does:This command runs the com.example.App class from the JAR file generated in the previous step.

**Expected Output:**

The output should display:

Hello World!

**Part B: Migrate the Application to Gradle**

In this part, you will create a Gradle project that contains the same Java application code, then build and run it using Gradle.

**Create a New Gradle Project:**

## Step1:Initialize Gradle in Your Project

### Open Terminal in IntelliJ IDEA

Make sure you're in the project directory:cd "D:\Idea Projects\MVNGRDLDEMO"

**Step2: Initialize the Gradle Project Using the Java Application Type:**

Execute the following command to migrate your Maven project to Gradle:

**gradle init --type java-application**

What it does: Gradle will generate a basic Java application project with a default project structure.

Adjust the Gradle Project to Use the Same Code

The Gradle project structure will look similar to this:

HelloMavenGradle/

├── build.gradle

├── settings.gradle

└── src

├── main

│ └── java

│ └── App.java

└── test

└── java

└── AppTest.java

## **Step 3: Review and Update**

Update the Gradle Build Script

**Open the build.gradle File in Your Editor:**

**Modify the application Block to Set the Correct Main Class:**

Change or add the following in the file:

plugins {

id 'application'

}

application {

// Update the mainClass to reflect the package structure mainClass = 'com.example.App'

}

**Build and Run the Gradle Application**

**Build the Project:**

gradle build

What it does: Compiles the source code, runs tests, and packages the application.

**ExpectedOutput:**

Look for a “BUILD SUCCESSFUL” message.

**Run the Application Using Gradle:**

gradle run

**What it does:** Uses the application plugin to run the main class defined in the build.gradle file.

**Expected Output:**

The output should display:

Hello World!