

Experiment 1: Working with Maven: Creating a Maven Project, Understanding the POM File, Dependency Management and Plugins.

Pom script:

```
<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>

  <!-- Project Coordinates -->

  <groupId>com.example</groupId>

  <artifactId>MyMavenApp</artifactId>

  <version>1.0-SNAPSHOT</version>

  <!-- Dependencies -->

  <dependencies>

    <!-- Example: JUnit for testing -->

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.13.2</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <!-- Build Configuration -->

  <build>

    <plugins>

      <!-- Maven Compiler Plugin -->

      <plugin>

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

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

        <version>3.8.1</version>

        <configuration>
```

```
        <source>11</source>

        <target>11</target>

    </configuration>

</plugin>

<!-- Maven Surefire Plugin (for running tests) -->

<plugin>

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

    <artifactId>maven-surefire-plugin</artifactId>

    <version>2.22.2</version>

</plugin>

</plugins>

</build>

</project>
```

Experiment 2: Working with Gradle: Setting Up a Gradle Project, Understanding Build Scripts (Groovy DSL), Dependency Management and Task Automation.

Groovy script:

```
plugins {  
    // Apply the Java plugin for compiling Java code  
    id 'java'  
    // Apply the application plugin to add support for building an application  
    id 'application'  
}  
  
group = 'com.example'  
version = '1.0'  
  
repositories {  
    // Use Maven Central for resolving dependencies.  
    mavenCentral()  
}  
  
dependencies {  
    // Define your dependencies. For example, JUnit for testing:  
    testImplementation 'junit:junit:4.13.2'  
}  
  
application {  
    // Define the main class for the application.  
    mainClass = 'com.example.App'  
}  
  
// A custom task example: printing a greeting  
task hello {  
    doLast {  
        println 'Hello, Gradle!'  
    }  
}
```

Experiment 3: Working with Gradle: Setting Up a Gradle Project, Understanding Build Scripts (Kotlin DSL), Dependency Management and Task Automation.

Kotlin script:

```
plugins {  
    // Apply the Java plugin for compiling Java code  
    java  
    // Apply the application plugin to add support for building an application  
    application  
}  
group = "com.example"  
version = "1.0"  
repositories {  
    mavenCentral()  
}  
dependencies {  
    // Define dependencies using Kotlin DSL syntax  
    testImplementation("junit:junit:4.13.2")  
}  
application {  
    // Set the main class for the application  
    mainClass.set("com.example.App")  
}  
// A custom task example using Kotlin DSL  
tasks.register("hello") {  
    doLast {  
        println("Hello, Gradle with Kotlin DSL!")  
    }  
}
```

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

Maven script:

```
<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>HelloMaven</artifactId>
  <version>1.0-SNAPSHOT</version>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.13.2</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>11</source>
          <target>11</target>
        </configuration>
      </plugin>
```

</plugins>

</build>

</project>

Groovy Script:

```
plugins {  
    id 'application'  
    id 'java'  
}  
application {  
    mainClass = 'com.example.App' // Correct way in Gradle 7+  
}  
repositories {  
    mavenCentral()  
}  
dependencies {  
    implementation 'org.apache.commons:commons-lang3:3.12.0'  
    testImplementation 'junit:junit:4.13.2'  
}
```