**Build Lifecycle  
Demonstrate the use of Maven lifecycle phases (clean, compile, test, package, install, deploy) by executing them on a sample project and documenting what happens in each phase.**

The Maven build lifecycle is a series of phases that are executed in a particular order to build, test, and deploy a project. The main lifecycle phases are:

**Clean**

The clean phase removes all files generated by the previous build. This phase is useful for cleaning up the project directory and ensuring that the build starts from a clean slate.

**Compile**

The compile phase compiles the source code of the project. This phase is responsible for transforming the Java source code into bytecode.

**Test**

The test phase runs the unit tests of the project. This phase is responsible for verifying that the code works as expected.

**Package**

The package phase packages the compiled code into a distributable format such as a JAR or WAR file.

**Install**

The install phase installs the packaged artifact into the local repository. This phase is useful for making the artifact available for other projects to use.

**Deploy**

The deploy phase deploys the packaged artifact to a remote repository. This phase is useful for making the artifact available to other teams or environments.

Here's an example of executing these phases on a sample project:

Let's create a new Maven project using the following command:

mvn archetype:generate

This will create a basic project structure with a pom.xml file.

Next, let's add some dependencies to the pom.xml file:

<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>my-project</artifactId>

<version>1.0</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

Now, let's execute the Maven lifecycle phases:

**Clean**

mvn clean

This will remove all files generated by the previous build.

**Compile**

mvn compile

This will compile the source code of the project.

**Test**

mvn test

This will run the unit tests of the project.

**Package**

mvn package

This will package the compiled code into a JAR file.

**Install**

mvn install

This will install the packaged artifact into the local repository.

**Deploy**

mvn deploy

This will deploy the packaged artifact to a remote repository.

Note that each phase builds upon the previous one, so executing a phase will also execute all the preceding phases. For example, executing mvn package will also execute the compile and test phases.