Task 1: 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.

1. Setting Up:

- Create a simple Maven project directory with a pom.xml file.
- Add basic project information like groupId, artifactId, and packaging (e.g., jar).

2. Phase Execution:

We'll use the mvn command followed by the specific lifecycle phase to execute each step.

clean:

- Run mvn clean.
- This phase removes all target directories and project build outputs from previous builds, cleaning up the workspace.

validate:

- Run mvn validate.
- This phase validates the project configuration in your pom.xml file. It checks for missing dependencies, plugins, or errors in the project structure.

compile:

- Run mvn compile.
- This phase compiles your project's source code (usually Java) into bytecode (.class files). These files are placed in the target/classes directory.

test:

- Run mvn test.
- This phase executes unit tests associated with your project. These tests verify the functionality of your code in isolation. The test results are displayed in the console.

package:

- Run mvn package.
- This phase packages the compiled code (.class files) along with dependencies and configuration files into a distributable artifact (e.g., JAR file). This artifact is placed in the target directory.

install:

- Run mvn install.

- This phase installs the packaged artifact (JAR) into your local Maven repository. This makes the artifact readily available for other projects that depend on it.

deploy:

- Run mvn deploy (**Note:** This usually requires remote repository configuration).
- This phase deploys the packaged artifact to a remote repository (e.g., company Nexus server). This allows other developers to access and use your project as a dependency.