K DEEPAK ASSIGNMENT-DAY-31

DAY-24-CORE JAVA

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.

Maven Lifecycle Demonstration:

Objective: Demonstrate the use of Maven lifecycle phases (clean, compile, test, package, install, deploy) by executing them on a sample project and documenting the actions taken in each phase.

Maven Lifecycle Phases Overview:

clean: This phase deletes all files generated by the previous builds to start fresh.

compile: This phase compiles the project's source code.

test: This phase runs the project's tests using a suitable framework like JUnit.

package: This phase packages the compiled code and resources into distributable formats such as JAR, WAR, or EAR.

install: This phase installs the packaged artifact into the local Maven repository for use as a dependency in other local projects.

deploy: This phase deploys the packaged artifact to a remote repository for use by other projects or developers.

Step-by-Step Demonstration:

Create a Sample Maven Project:

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

Begin by generating a new Maven project with the following command:

This creates a simple Maven project with a basic structure and a sample Java class.

Navigate to the Project Directory:

```
cd my-maven-project
```

Execute Maven Lifecycle Phases:

Run each Maven lifecycle phase sequentially with the following commands:

clean: Removes all files generated by previous builds.

```
    mvn clean
```

compile: Compiles the project's source code.

```
    mvn compile
```

test: Executes the project's tests.

```
    mvn test
```

package: Packages the compiled code into a JAR file.

```
mvn package
```

install: Installs the packaged artifact into the local Maven repository.

```
mvn install
```

deploy: Deploys the packaged artifact to a remote repository. This phase requires remote repository configuration, so it may not perform any action if not configured.

```
mvn deploy
```

Review the Output:

After executing each command, Maven will display the progress of the build process, including any relevant output or errors. The target directory will contain the generated artifacts (e.g., JAR file) after the package phase.

By following these steps, you can observe how Maven handles the build process, from cleaning previous builds to compiling, testing, packaging, installing, and deploying the project artifacts. Each phase plays a vital role in the overall build lifecycle, ensuring a streamlined and efficient software development process.