Assignment – 25

Prakash Manikanta Irrinki

Prakashnaidu9494@gmail.com

Task 1:
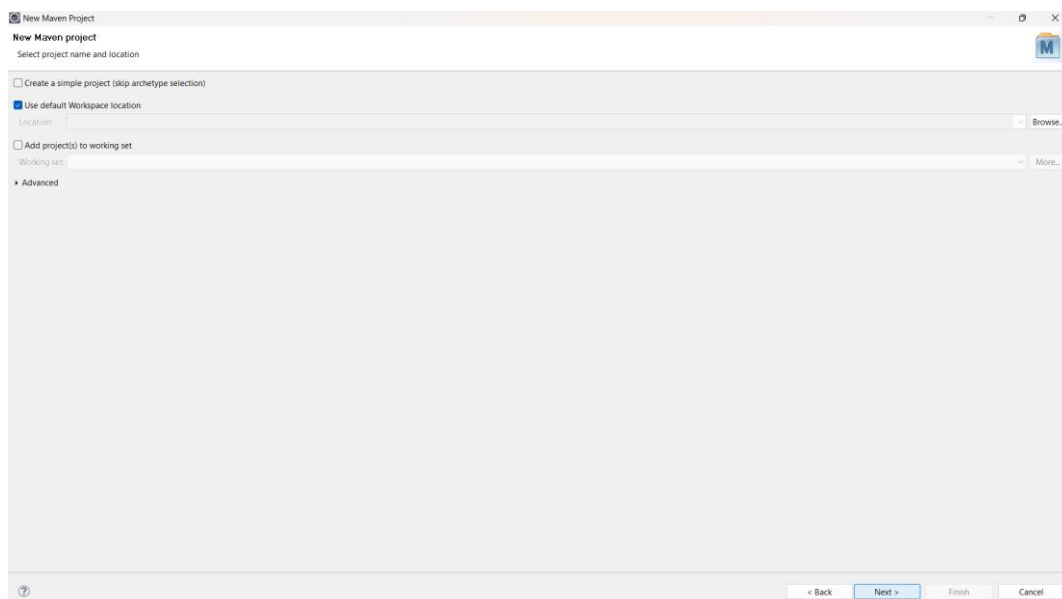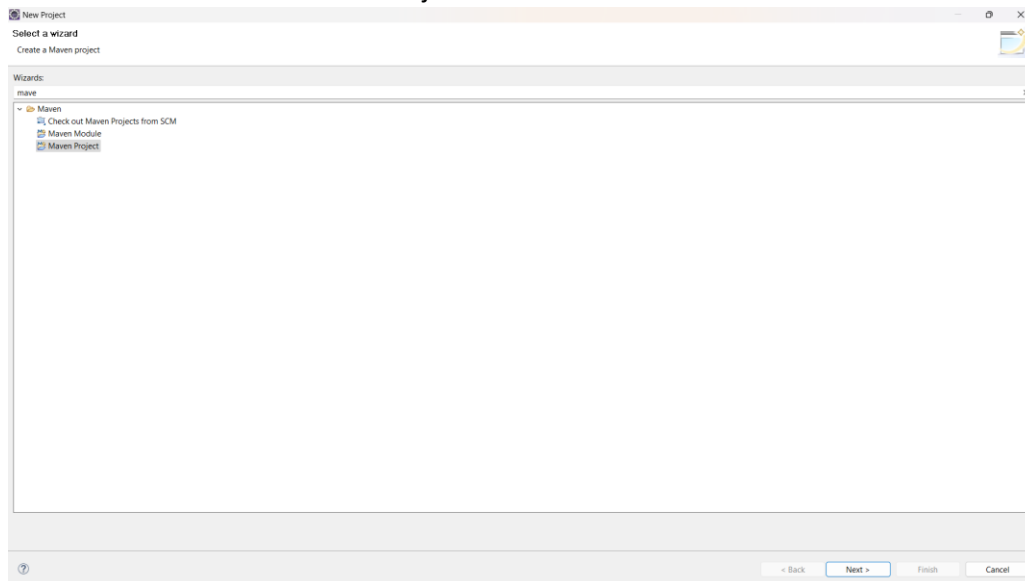
Build Lifecycle:

==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.==

Step 1: Create a Maven Project:

Right click

New

Maven Project

New Maven Project

New Maven project
Select an Archetype

Catalog: Internal ▾ Configure...

All Catalogs
Internal
Default Local
Maven Central

Filter:

Group Id

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-j2ee-simple	1.0
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1
org.apache.maven.archetypes	maven-archetype-site-simple	1.1
org.apache.maven.archetypes	maven-archetype-webapp	1.0

An archetype which contains a sample Maven project.

☑ Show the last version of Archetype only   ☐ Include snapshot archetypes   Add Archetype...

▸ Advanced

< Back   Next >   Finish   Cancel

New Maven Project

New Maven project
Select an Archetype

Catalog: Internal ▾ Configure...

Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-archetype	1.0
org.apache.maven.archetypes	maven-archetype-j2ee-simple	1.0
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1
org.apache.maven.archetypes	maven-archetype-site-simple	1.1
org.apache.maven.archetypes	maven-archetype-webapp	1.0

An archetype which contains a sample Maven project.

☑ Show the last version of Archetype only   ☐ Include snapshot archetypes   Add Archetype...

▸ Advanced

< Back   Next >   Finish   Cancel

New Maven Project

New Maven project
Specify Archetype parameters

Group Id: com.rps.assignment
Artifact Id: mavenproject
Version: 0.0.1-SNAPSHOT ▾
Package: com.rps.assignment.mavenproject
☑ run archetype generation interactively

Properties available from archetype:

Name	Value

Add...
Remove

▸ Advanced

< Back   Next >   Finish   Cancel

Enter Group Id
Enter Artifact Id
        Finish

Step 2: Add Java Source Code

Navigate into the 'Mavemproject' directory and create a simple Java class and a corresponding JUnit test class.

src/main/java/com/rps/assignment/mavemproject/App.java

```java
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello, Maven!" );
    }

}
```

src/main/java/com/rps/assignment/mavemproject/Test.java

```java
import junit.framework.TestCase;

public class AppTest extends TestCase {

    public AppTest(String testName) {
        assertTrue(true);
    }

}
```

Step 3: Maven Lifecycle Phases
Now, let's execute each Maven lifecycle phase and observe what happens.

1. Clean phase:
This phase cleans up artifacts created by previous builds.

Command:
C:\Users\praka\eclipse-workspace\mavemproject>mvn clean

Output:

[INFO] Scanning for projects...
[INFO]
[INFO] -----------------< com.rps.assignment:mavemproject >------------------
[INFO] Building mavemproject 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[jar]---------------------------------

[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ mavemproject ---
[INFO] Deleting C:\Users\praka\eclipse-workspace\mavemproject\target
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 0.431 s
[INFO] Finished at: 2024-06-16T15:47:52+05:30
[INFO] --

2. Compile phase

This phase compiles the source code of the project.

Command:

C:\Users\praka\eclipse-workspace\mavemproject>mvn compile

Output:

[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< com.rps.assignment:mavemproject >-------------------
[INFO] Building mavemproject 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-
workspace\mavemproject\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mavemproject ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 1.411 s
[INFO] Finished at: 2024-06-16T15:50:31+05:30
[INFO] --

The Java source code (App.java) is compiled into byte code (App.class) in the target/classes
directory.

3. Test phase:

This phase runs tests using a suitable unit testing framework (like JUnit).

Command:

C:\Users\praka\eclipse-workspace\mavemproject>mvn test

Output:

[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< com.rps.assignment:mavemproject >------------------
[INFO] Building mavemproject 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-workspace\mavemproject\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mavemproject ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-workspace\mavemproject\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mavemproject ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ mavemproject ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running com.rps.assignment.mavemproject.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 s -- in com.rps.assignment.mavemproject.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 2.004 s
[INFO] Finished at: 2024-06-16T15:58:22+05:30

[INFO] --

The test AppTest.java is executed successfully (in this case, it just asserts true).

4. Package phase:

This phase packages the compiled code into a distributable format, such as a JAR.

Command:
C:\Users\praka\eclipse-workspace\mavemproject>mvn package

Output:

[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< com.rps.assignment:mavemproject >-------------------
[INFO] Building mavemproject 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-
workspace\mavemproject\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mavemproject ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-
workspace\mavemproject\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mavemproject ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ mavemproject ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running com.rps.assignment.mavemproject.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 s -- in
com.rps.assignment.mavemproject.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ mavemproject ---
[INFO] Building jar: C:\Users\praka\eclipse-workspace\mavemproject\target\mavemproject-0.0.1-
SNAPSHOT.jar

[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 2.358 s
[INFO] Finished at: 2024-06-16T16:06:38+05:30
[INFO] --
The compiled classes and resources are packaged into a JAR file (mavemproject-0.0.1-SNAPSHOT.jar) in the target directory.

5. Install phase:

This phase installs the packaged JAR file into the local Maven repository.

Command:
C:\Users\praka\eclipse-workspace\mavemproject>mvn install

Output:

[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< com.rps.assignment:mavemproject >------------------
[INFO] Building mavemproject 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-workspace\mavemproject\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mavemproject ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ mavemproject ---
[INFO] skip non existing resourceDirectory C:\Users\praka\eclipse-workspace\mavemproject\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mavemproject ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ mavemproject ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running com.rps.assignment.mavemproject.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.046 s -- in com.rps.assignment.mavemproject.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ mavemproject ---
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ mavemproject ---
[INFO] Installing C:\Users\praka\eclipse-workspace\mavemproject\pom.xml to
C:\Users\praka\.m2\repository\com\rps\assignment\mavemproject\0.0.1-
SNAPSHOT\mavemproject-0.0.1-SNAPSHOT.pom
[INFO] Installing C:\Users\praka\eclipse-workspace\mavemproject\target\mavemproject-0.0.1-
SNAPSHOT.jar to C:\Users\praka\.m2\repository\com\rps\assignment\mavemproject\0.0.1-
SNAPSHOT\mavemproject-0.0.1-SNAPSHOT.jar
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 2.070 s
[INFO] Finished at: 2024-06-16T16:13:10+05:30
[INFO] --

The JAR file (mavemproject-0.0.1-SNAPSHOT.jar) and its corresponding POM file are installed in the local Maven repository (~/.m2/repository).

6. Deploy phase:

This phase copies the final package to a remote repository for sharing with other developers and projects.

Note: For deploy to function, you typically need to configure a remote repository. Since this setup is often specific to each organization, I'll skip executing this phase here.

Summary:

➔ **Clean**: Deletes the target directory.
➔ **Compile**: Compiles main source code to target/classes.
➔ **Test**: Runs tests from src/test directory.
➔ **Package**: Packages compiled code into a JAR