**Lab Exercise 10**

**Aim:** Build and Deploy a sample Maven project

Create a complete Maven Project and use it in some way in another java project.

**Project 1: Maven Project**

**Java Class source code:** HelloWorld.java

package lab2;

public class HelloWorld {

public String function1()

{

return "Nishkarsh Raj Is Awesome!!";

}

public static void main(String args[])

{

System.out.println("Hello!");

}

}

**JUnit Test case:** Test1.java

package lab2;

import static org.junit.Assert.\*;

import org.junit.Test;

public class Test1 {

@Test

public void test() {

//fail("Not yet implemented");

}

}

**POM 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>upesdevops</groupId>

<artifactId>lab2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source> 9 </maven.compiler.source>

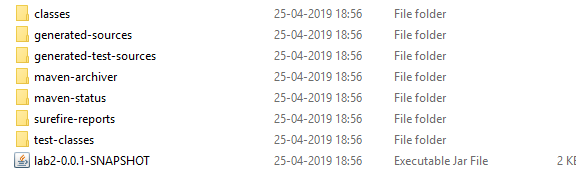
<maven.compiler.target> 9 </maven.compiler.target>

</properties>

</project>

**Building the Maven project:**

1. Mvn clean
2. Mvn compile
3. Mvn test
4. Mvn install



**Project 2: Java Project**

* Configure the build path
* Add External JAR: The target file created in project 1.
* Create a new Java class: Runner.java

**Source Code:**

package lab2runner;

import lab2.\*;

public class Runner extends HelloWorld {

public static void main(String args[])

{

HelloWorld obj = new HelloWorld();

obj.function1();

}

}

