

## Module-8 Modular Programming

### Problem Statement 1:

Set up a new Maven project and add external dependencies using Maven Central. Understand the role of the pom.xml file in dependency management.

### Problem Statement 2:

Explore Maven's transitive dependency resolution. Create a project with multiple dependencies and observe how Maven resolves and downloads them.

A demo maven project. Pom.xml

```
<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>demo</groupId>
  <artifactId>MavenDemoProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>

    <!-- JUnit Dependency -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter</artifactId>
      <version>5.10.0</version>
      <scope>test</scope>
    </dependency>

    <!-- Mockito Dependency -->
    <dependency>
      <groupId>org.mockito</groupId>
      <artifactId>mockito-core</artifactId>
      <version>5.5.0</version>
      <scope>test</scope>
    </dependency>

    <!-- Mockito JUnit Integration -->
    <dependency>
      <groupId>org.mockito</groupId>
      <artifactId>mockito-junit-jupiter</artifactId>
      <version>5.5.0</version>
      <scope>test</scope>
    </dependency>

  </dependencies>
</project>
```