**Software Design and Engineering**

**Lab Document**

|  |  |
| --- | --- |
| **High Level Purpose Statement:** | I want to design and implement an experiment that tests how to use Maven effectively for building, managing dependencies, and packaging a Java project into a deployable artifact. Additionally, I want to explore Maven’s plugin system for automating tasks such as testing and creating documentation. |
| **Experimental Design:** | Select or create a Java project that utilizes at least two external libraries as dependencies. For this experiment, I will start with a project that includes functionality like file manipulation and logging.  Install Maven if it is not installed on my system. Initialize the project as a Maven project by using IntelliJ’s built-in tools for converting a standard Java project into a Maven project.  Write unit tests for the project (e.g., using JUnit) and configure Maven’s Surefire Plugin to automate test execution. Use Maven commands such as mvn compile, mvn test, and mvn package to build the project and verify its functionality.  Configure Maven to package the project into a JAR file using the Maven Assembly or Shade Plugin to include all dependencies. Test the JAR file by running it on another computer to ensure it works independently.  Investigate other useful Maven plugins, such as those for generating project documentation (e.g., Apache Maven Site Plugin) or managing versioning.  Create a step-by-step guide on setting up and using Maven for this project, including troubleshooting common issues. |
| **Resources Available:** | Official Maven Documentation. IntelliJ IDEA Documentation. Online tutorials and videos on Maven and its plugins. Maven community forums and Stack Overflow. |
| **Time Estimate:** | I estimate spending around 12 hours on this experiment:  3 hours setting up the Java project and integrating dependencies  3 hours learning Maven basics and setting up the pom.xml file  3 hours experimenting with build, test automation, and packaging  3 hours exploring and documenting the use of Maven plugins |
| **Experiment Notes:** | Record key challenges encountered while setting up Maven and plugins. Document all commands, pom.xml configurations, and errors encountered, along with their resolutions. |
| **Results:** | Summarize how Maven simplified (or complicated) the build and deployment process. Identify the most useful Maven features and plugins based on this experiment. |
| **Consequences for the Future:** | Use Maven as a standard build and dependency management tool for future Java projects. Incorporate Maven’s plugin ecosystem to streamline workflows like testing, packaging, and documentation. Share insights with peers or incorporate them into a tutorial for others starting with Maven. |