JAVA-MVN: Java with Maven

Course Overview

A comprehensive course outline for JAVA-MVN: Java with Maven balances core Java programming fundamentals with modern build automation using Apache Maven. This syllabus is designed to take a developer from setting up an environment to deploying enterprise-ready, multi-module applications.

Course Description

This course bridges the gap between core programming and modern software delivery.Participants will master the Java language alongside Apache Maven, the industry-standard build automation tool.You will learn to write clean, object-oriented code while automating repetitive developer tasks. These tasks include downloading third-party libraries, compiling code, running test suites, and packaging software.By the end of this training, you will transition from writing isolated scripts to building, testing, and shipping production-ready enterprise applications.

Learning Curriculum

Module 1

Environment Setup & Java Fundamentals

JDK and Maven Installation: Setting up Java (JDK 21 or later) and Apache Maven on Windows, macOS, or Linux.Environment Variables: Configuring JAVA_HOME, MAVEN_HOME, and updating system path variables.Java Syntax Refresher: Variables, data types, control flow statements, and arrays.Object-Oriented Programming (OOP): Classes, inheritance, polymorphism, encapsulation, and interfaces.Java Core APIs: Working with Collections framework, Exception Handling, and File I/O

Module 2

Introduction to Maven & Project Layout

What is a Build Tool?: Understanding build automation, compiling, and the challenges of manual dependency management.Maven vs. Competitors: Comparing Maven with Apache Ant and Gradle.Convention Over Configuration: Exploring Maven's standardized folder layouts.Standard Directory Structure:src/main/java for source codesrc/main/resources for configuration propertiessrc/test/java for test casesMaven Archetypes: Generating project skeletons using command-line templates (mvn archetype:generate

Module 3

The Project Object Model (POM)

Anatomy of pom.xml: Understanding the core configuration file of any Maven project.Maven Coordinates: Defining groupId, artifactId, and version (GAV).Properties & Metadata: Managing project-wide variables and encodings.Super POM & Effective POM: Understanding how inheritance works globally in Maven configs.

Module 4

Dependency Management

Declaring Dependencies: Adding external libraries (e.g., Log4j, Jackson) to your project.Maven Repositories: How Maven searches Local (~/.m2), Central, and Remote repositories.Dependency Scopes: Controlling classpath visibility using compile, provided, runtime, test, and system.Transitive Dependencies: Navigating dependency trees and resolving version conflicts using exclusions

Module 5

Build Lifecycles, Phases, and Goals

The Three Core Lifecycles: Exploring default, clean, and site lifecycles.Default Lifecycle Phases: Walking through validate, compile, test, package, verify, install, and deploy.Essential Maven Commands: Executing terminal operations like mvn clean install and mvn package.Plugins and Goals: How plugins bind to lifecycle phases to execute tasks

Module 6

Testing and Code Quality Integration

Unit Testing with Maven: Integrating JUnit 5 and TestNG frameworks.Surefire Plugin: Running, selective filtering, and skipping test suites from the command line.Failsafe Plugin: Setting up integration testing isolated from unit testing.Code Coverage: Integrating the JaCoCo plugin to analyze code testing health

Module 7

Advanced Maven Techniques

Executable Heavy Artifacts: Creating fat/uber JARs using the Maven Shade or Assembly plugins.Build Profiles: Customizing configurations for dev, test, and prod environments.Multi-Module Projects: Managing complex applications via a Parent POM and Child sub-modules.IDE Integration: Managing Maven inside modern IDEs like IntelliJ IDEA and Eclipse

© 2026 Proasia Upskilling Sdn Bhd | www.proasia.my