

## ATC-J8M: Java 8 With Maven

Course Code: ATC-J8M

Duration: 5 days

Instructor-led Training (ILT) | Virtual Instructor-led Training (VILT)

### OVERVIEW

Intensive and hands-on, the course emphasizes becoming productive quickly as a Java application developer. This course quickly covers the Java language syntax and then moves into the object-oriented features of the language. As a participant you will be able to create basic applications using the Java 8 programming language.

The participants will also be able to describe object-oriented concepts, object-oriented programming (OOP) concepts, and Java platforms and technologies. This course will use Eclipse as an IDE.

### SKILLS COVERED

Participants will learn how to:

- Write, compile and execute Java programs
- Build robust applications using Java's object-oriented features
- Create robust applications using Java class libraries

### WHO SHOULD ATTEND?

This course is meant for programmers who wish to move to object-oriented programming using Java. This course is designed for participants who are looking for an entry into an application development or a software project management career using Java technologies.

### PREREQUISITES

The course assumes knowledge of programming.

### MODULES

#### Module 1: Programming Concepts

- What is Programming
- Components of a Programming Language
- Programming Paradigms
- What is Java
- The Object-Oriented Paradigm and Java

#### Module 2: Writing Java Programs

- What is a Java Program
- Writing the Source Code
- Compiling the Source Code
- Running the Compiled Code
- Installing and configuring Eclipse IDE
- Downloading and Installing Maven
- Testing Maven Installation

#### Module 3: Data Types

- What is a Data Types
- What is an Identifier
- Data Types in Java
- Primitive Data Types in Java
- Special Floating-Point Numbers
- Rounding Modes
- Number Objects
- String Objects
- Format data for output

#### Module 4: Operators

- What is an Operator
- Assignment Operator
- Arithmetic Operators
- String Concatenation Operator
- Relational Operators
- Boolean Logical Operators

**Module 5: Statements**

- What is a Statement and Type of Statements
- A Block Statement
- The if-else Statement
- The switch Statement
- Strings in Switch
- The for Statement
- The for-each Statement
- The while Statement
- The do-while Statement
- The break Statement
- The continue Statement
- Diamond Syntax

**Module 6: Classes and Objects**

- What is a Class
- Declaring Fields in a Class
- Creating Instances of a Class
- The null Reference Type
- Using Dot Notation to Access Fields of a Class
- Default Initialization of Fields
- Access Level Modifiers for a Class
- Import Declaration
- Declaring Methods of a Class
- Local Variables
- Instance Method and Class Method
- Invoking a Method
- Access Level for Class Members

**Module 7: The Object and Objects Classes**

- The Object Class
- What is the Class of an Object
- Computing Hash Code of an Object
- Comparing Objects for Equality
- String Representation of an Object
- Cloning Object
- Finalizing an Object
- Immutable Objects

**Module 8: Java Package**

- Package declaration
- Sub-Packages in Java
- Naming Convention
- Importing Java Package
- Access Modifier
- Java Archive (JAR) Files

**Module 9: Wrapper Classes**

- Wrapper Classes
- Autoboxing and Unboxing
- Beware of Null Values
- Overloaded Methods
- Comparison Operators
- Collections

**Module 10: Exception Handling**

- What is an Exception
- Using a try-catch Block
- Exception Class Hierarchy
- Arranging Multiple catch Blocks
- Throwing an Exception
- Creating an Exception Class
- The finally Block
- The try-with-resources Block
- Multi-Catch Exceptions
- Re-throwing Exceptions
- Try with Resources
- AutoCloseable Interface

**Module 11: Assertions**

- What is an Assertions
- Testing Assertions
- Enabling / Disabling Assertions
- Using Assertions

**Module 12: Strings**

- What is a String
- String Literals
- Creating String Objects
- String Operators
- Comparing Two Strings

- StringBuilder and StringBuffer

### Module 13: Dates and Times

- The Date-Time API
- Calendar Systems
- Clocks and Period Packages
- Period between Two Dates and Times
- Partial
- Formatting Dates and Times
- Parsing Dates and Times

### Module 14: Formatting Data

- Wrapper Classes
- Autoboxing and Unboxing
- Beware of Null Values
- Overloaded Methods
- Comparison Operators

### Module 15: Arrays

- What is an Array
- Arrays are Objects
- Accessing Array Elements
- Length of an Array
- Multi-Dimensional Arrays
- Enhanced for Loop for Arrays
- Runtime Array Bounds Check

### Module 16: Inheritance

- What is Inheritance
- The instance of Operator
- Binding
- Method Overriding and Method Overloading
- Method Hiding and Field Hiding
- Disabling Inheritance

### Module 17: Interfaces

- What is an Interface
- Implementing an Interface
- Implementing Multiple Interfaces

- Interface Inheritance
- The instanceof Operator
- Marker Interfaces
- Functional Interfaces
- Polymorphism – One Object, Many Views
- Dynamic Binding and Interfaces

### Module 18: Core Collection Classes

- The Collections Framework
- The Set Interface
- Set Implementation Classes
- The List Interface
- List Implementation Classes
- The Map Interface
- Map Implementation Classes

### Module 19: Pluggable Annotation Processing API

- Common Annotations and Role Based Annotations
- Generated
- Resource
- Resources
- PostConstruct
- PreDestroy
- DeclareRoles
- RolesAllowed
- PermitAll
- DenyAll

### Module 20: Java API for XML Based Web Services – 2.0

- JAXB 2.0
- Web Services Metadata
- Streaming API for XML
- XML Digital Signature
- Java Class File Specification Update
- Java Compiler API

### Module 21: JDBC 4.0

- Setting up MySQL Database

- Auto loading of Driver by JVM
- Standard Connection Factory Management
- Enhanced Connection and Statement Interface
- Inserting Data
- New interface RowID to support the ROWID data type
- Updating Data
- Deleting Data
- Use of Annotated SQL queries
- Creating prepared statements
- Calling stored procedure

### Module 22: Lambda Expressions

- Single method Interface
- Anonymous Interface Implementation
- Lambda Type Inference
- Lambda Parameters
- Lambda Function Body
- Returning a Value
- Lambdas as Objects
- Variable Capture
- Method References as Lambdas

### Module 23: Java Stream API for Bulk Data Operations on Collections

- Introduction
- Stream Creation
- Referencing a Stream
- Stream Pipeline
- Lazy Invocation
- Order of Execution
- Stream Reduction
- Parallel Streams

### Module 24: Java Date / Time API

- Issues with existing Date / Time APIs
- LocalDate, LocalTime and LocalDateTime
- Using ZonedDateTime API
- Using Period and Duration

- Compatibility with Date and Calendar
- Date and Time Formatting

### Module 25: Concurrency API Improvements

- The Concurrency API
- Semaphore and Other Synchronizers
- Concurrent Collections
- Atomic Operations
- Executor and ExecutorService
- Thread Pools
- Parallel Processing

### Module 26: Reflection

- Uses for Meta-Data
- The Reflection API
- The ClassClass
- The java.lang.reflect Package
- Reading Type Information
- Navigating Inheritance Trees
- Dynamic Instantiation
- Dynamic Invocation
- Reflecting on Generics

### Module 27: Getting Started With Maven

- Terminology and Basic Concepts
- Artifacts
- Lifecycle
- Plugins
- Running Maven
- Running Maven from Eclipse
- Common Goals

### Module 28: Commonly Used Plugins

- Maven Plugins
- Declaring and Configuring Plugins
- Running the Plugin
- Binding a Plugin Goal to the Lifecycle
- Maven Surefire Test Plugin
- Failsafe Plugin
- Site Plugin
- JavaDoc Plugin

- PMD Plugin

**END OF PAGE**

### **Module 29: Multi-Module Builds**

- Introduction
- The Reactor
- Reactor Sorting
- Multi-Module Build by Example

### **Module 30: POM Projects**

- Project Object Model (POM)
- The overall POM structure
- Storing POM
- What is Maven Plugin
- Example of Using a Plugin
- Create a Custom Plugin
- Create a Custom Plugin (cont.)
- Plugin Management

### **Module 31: Creating Archetypes**

- Introduction to Maven Archetypes
- Introduction to Maven Archetypes (cont.)
- Using Interactive Mode to generate Goal
- Common Maven Archetypes

### **Module 32: Repository Management**

- Maven's Approach to Artifacts
- Publishing Artifacts
- Summary of Maven's Artifact Handling
- Repository
- Repository Manager
- Proxy Remote Repositories
- Types of Artifacts
- Release Artifacts
- Snapshot Artifacts
- Reasons to Use a Repository Manager
- Repository Coordinates
- Addressing Resources in a Repository