



Intermediate Java

Course Overview:

The **Intermediate Java training course** examines common language features and APIs required to develop complex stand-alone Java applications. Intermediate Java builds on the [Introduction to Java](#) course and examines things like advanced OO concepts, threaded programming, logging, custom annotations, and new features in Java 8.

Productivity Objectives:

After this course, you will be able to:

- Identify and work with new language features introduced in Java SE 7 and SE 8
- Create and work with complex Lambda expressions
- Create custom annotations
- Work with the new `java.time` API
- Use the `java.util.concurrent` package for threading

Course Duration: This course will be delivered in 3 days.

Course Outline:

- Review of core Java Syntax
 - Language syntax
 - Exceptions
 - Packages
 - Collection APIs
- Advanced OO concepts
 - Class partitioning for deployment

- Inner Classes
- Working with the Logging library
 - What is the logging library
 - Configuring the logger
 - Using a logger
 - Logging exceptions
 - java.util.logging v. Log4J
- Working with java.time API
 - Continuous vs. Human
 - Calculating time
 - Using time-zones
 - Parsing and formatting
 - Supporting legacy date/time
- Advanced Java 7 and Java 8 Features
 - Creating custom Generics
 - Customizing the Collections API through custom algorithms and data structures
 - Advanced Lambda Expressions and parallel operations
 - Functional Interfaces
 - Default methods
- Creating custom annotations
 - Defining an annotation
 - Interpreting an annotation at run-time
 - Generating dynamic code based on a run-time annotation
 - Loading dynamic class
 - Using repeating annotations
- Advanced Java Threads
 - Introduction to Concurrent libraries
 - Collection modifications and Concurrent Collections
 - Task Scheduling Framework
 - Synchronizers
 - Locks
 - Atomic variables
 - Timing
 - Concurrent accumulators
 - Fork-Join