

# Advanced Java – Course Outline

---

## 1 Duration

3 days

## 2 Objectives

This course provides thorough coverage of the advanced topics in Java. New concepts such as the use of lambda expressions, stream API, enhanced concurrency are covered in detail.

1. Understand internals of Java VM, Memory Management and Garbage Collection
2. Recap Java Core Concepts and Java Collection Framework
3. Understand Java Inner Classes, Reflection API, Generics and Annotations
4. Understand functional programming and lambda expressions
5. Understand Stream API and its usage
6. Understand concurrency concepts and develop multi-threaded applications

Note: Java 11 and Eclipse latest version will be used as development environment

## 3 Audience

This course is for people who wanted to learn advanced Java concepts and new features

## 4 Pre-requisite

- Java SE level programming proficiency or completion of an introductory Java course

## 5 Hardware & Network Requirements

- All participants to have individual desktops/laptops with 4GB RAM (8 GB recommended)
- High-speed internet connection

## 6 Software Requirements

- Windows 64 Bit or Linux Ubuntu or Mac
- JDK 11, Eclipse latest version

## 7 Outline

### 7.1 Day 1

#### Module-1: Java Platform Internals

- Java Platform Overview
- Java VM Architecture
- Class Loader Overview
- Memory Management Overview
- Garbage Collection Overview
- VM startup arguments
- Monitoring and Management

#### Module-2: Recap Core Java and Collection Framework

- Recap Java Basics and OOPs concepts
- Recap Java Collection Framework
- Collection Data Structures - List, Set, Map, Queue
- Working with Collections
- Utility functions in Collections

#### Module-3: Generics

- Generics Overview
- Parameterized arguments and usage of wild cards
- Using class / interface level, Constructor level, Method Generic type declaration and accessing them
- Applying Generics on Collection Framework

#### Module-4: Annotations

- Types of Annotations – Single, Normal, Marker
- Specifying Targets – CLASS, METHOD, etc
- Specifying Retention Policy – SOURCE, CLASS, RUNTIME
- Understanding pre-defined annotations - @Override, @SuppressWarnings, @Deprecated
- Defining custom annotations
- Annotation processing

## 7.2 Day 2

### Module-5: Lambda Expressions

- Fundamentals of Functional Programming
- Lambda Expressions Overview
- Functional Interfaces
- Method References
- Type Inference

### Module-6: Stream API - Basics

- Stream API Overview
- Stream Operations and Patterns
- Filtering
- Mapping
- Finding and Matching
- Reducing
- minBy/maxBy

### Module-7: Stream API - Advanced

- Collectors Overview
- Grouping and partitioning
- Collecting to lists and sets
- Collection operations
- Aggregation and Reduction
- Parallelizing Stream Processing
- Stream Optimization

### Module-8: Enhanced Collections with Lambdas and other Java 8 features

- Default Methods
- Optional Data Types
- Enhancements in the existing Collection Framework with Lambdas and Streams
- Overview of other Java 8/11 features
- Lab exercises to create sample CRUD application with enhanced Collection Framework

## 7.3 Day 3

### Module-9: Java 11 features

- Java11 New Features:
- Module System
- String API Changes
- New File Methods
- Local Variable Syntax
- New HTTP client
- Java 11 Usecases

### Module-10: Multi-threading / Concurrency Overview

- Process vs Threads
- Thread Lifecycle
- Thread creation and execution
- Thread Constructs - wait, notify, notifyall, interrupt, sleep, join, volatile, yield
- Thread Priority
- Daemon threads
- Thread Groups

### Module-11: Concurrent Programming with Synchronization

- Race Condition and Deadlock
- Thread Safety - Atomicity and Locking
- Sharing Objects - Visibility, Publication, Thread confinement and Immutability
- Designing a thread-safe class and applying synchronization policies
- Synchronized containers, Concurrent containers, Blocking queues, Synchronizers

### Module-12: Concurrent Programming with Thread Pools

- Drawbacks of traditional multi-threading
- Concept of Resource Pooling leading to ThreadPools
- ExecutorService APIs
- Configuring Threadpools
- Submitting Runnable task
- Submitting Callable task and Future object
- Exception handling in thread pool
- Application of thread pool
- Working with concurrent collections – Blocking and non-blocking operations