

Java 8 Performance and Tuning (TT3110-J8)

Modality: Virtual Classroom

Duration: 2 Days

SATV Value:

CLC:

NATU:

SUBSCRIPTION: No

About this course:

In this course, you will learn the latest coding skills and techniques in Java 8 to develop benchmarks. You will also learn how to improve application and code performance, optimize your code, and much more.

The average salary of a Java Developer is **\$90,992** per year.

Course Objective:

- Metrics for measuring software performance
- Effect of OO design on software performance
- Types of benchmarks and the criteria that should be considered when constructing a benchmark plan
- Explore the most useful targets for profiling, and the most common tools/techniques for profiling
- Two strategies for improving performance as a result of profiling data
- Five most common problem areas for good performance with Java
- Use the JDK to collect runtime profiling data
- Read the profiling data generated by the JDK to detect performance bottlenecks
- Instrument your own code to collect method execution time data
- Learn code optimization techniques relating to object management, exceptions, threads, and serialization
- JVM Architecture from the perspective of performance
- Work with Java language features that can impact performance
- Optimizing data structures in Java
- Choose the correct collection for the task
- Leverage the built in collections algorithms to enhance your code performance and security
- Examine the many Java 8 features to understand their impact on performance

Audience:

Experienced Java developers who want to take their core Java skills and bring them to an advanced level.

Prerequisite:

At least six months of prior hands-on development experience working with Java is recommended

Course Outline:

Module 1. Writing High Performance Applications

- Memory Management Issues
- CPU Performance Issues
- Threading Issues
- Profiling and Benchmarking
- Java Microbenchmarking Harness (JMH)
- Code Optimization Techniques
- Design Optimization Techniques

Module 2. Effective Java

- Creating and Destroying Objects
- Factory Methods
- Impact of Finalizers
- Classes and Interfaces
- Immutability
- Composition vs. Inheritance
- Exceptions
- Threading Constructs to Avoid

Module 3. Data Structures

- Efficient Strings and Arrays
- Efficient Use of Collections
- Choosing a Collection
- Tuning Collection Constructors

Module 4. Performance and Java 8

- Performance Across Java Versions
- Impact of Java 8 Concurrency Updates
- Tiered Compilation
- Nashorn versus Java 7's JavaScript Engine
- Lambda Expressions versus Inner Classes
- Impact of Streams versus Collections
- Date/Time Classes
- I/O Stream Recommendations
- Encoding/Decoding
- Strings and String Interning