|  |  |
| --- | --- |
| **Course Outline Template** | |
| **Course Name** | **Tuning Java applications using Java Performance Tuning** |
| **Course Duration** | 5 Day (20 Hours) |
| **Course Timings** | 2:00PM to 6:00PM |
| **Course Start Date** | 1st Feb 2021 |
| **Course End Date** | 5th Feb 2021 |
| **Mode of Training** | Webinar |
| **Course Pre-requisites** | * Good understanding and working knowledge on Java * Working knowledge with Java IDE (Eclipse, STS or Intellij) |
| **Course Objectives** | Monitor operating system performance on Solaris, Linux, and Windows  Describe basic principles of performance  Describe the operation of generational garbage collection  List the garbage collectors available in Java including the G1 collector  Monitor performance at the JVM and application level  Monitor and analyze Java application performance using Java Mission Control and Flight Recorder  Profile the performance of a Java application  Tune garbage collection in a Java application  Apply basic performance tuning principles to a Java application |
| **Agenda - Day 1** | **Intro to Performance Tuning and JVM Architecture**   * Introduction to Performance Tuning * Java Platform Overview * JVM Architecture and Internals * Class Loader Subsystem * Runtime Data Area - Heap and Stack * Execution Engine - Interpreter, JIT Compiler, Garbage Collectors * Java Memory Management * Garbage Collection |
| **Agenda – Day 2** | **Performance Bottlenecks, Analysis and Monitoring Tools**   * Potential Performance Bottlenecks   + Memory Leaks   + High CPU Utilization   + Thread Concurrency Issues   + Garbage Collection Overhead   + Network Latency/Timeouts * Management and Monitoring Tools Overview   + Java VisualVM   + JConsole   + Java Mission Control * Detecting Memory Leaks - Heap Dump Analysis   Detecting reasons for High CPU Utilization |
| **Agenda – Day 3** | **Detecting Performance Bottlenecks and Profiling Tools**   * Profiling   + Memory Profiling   + CPU Profiling * Profiling Tools   + JProfiler   + YourKit   + Java Mission Control * Detecting Garbage Collection Overhead - GC Log Analyzer * Detecting Concurrency Issues - Thread Dump Analysis |
| **Agenda – Day 4** | **Tuning of Java Applications**   * JVM Tuning   + Memory Tuning   + GC Tuning   + JIT Tuning * Code Optimization * Caching * Load Balancing * Distributed Computing |
| **Agenda – Day 5** | **Tuning Java EE Applications**   * Measuring Java Enterprise Application performance * Analyzing the performance bottlenecks * Tuning the performance with different optimization techniques * Web/App Server and Middleware Tuning * Operating System / Network Tuning * Best Practices |