Advanced Concurrency

# Duration

* 3 Days

# Objectives

At end of this workshop, participants will able to :

* Get an overall understanding of various concurrency techniques and methods available in Java
* Design and build concurrent applications more effectively
* Get knowledge on performance tuning, troubleshooting and debugging of concurrent programs

# Audience

Java Developers, Leads and Architects

# Pre-requisite

* Expertise in Java programming language
* Good knowledge on Multi Threading concepts

# Hardware & Network Requirements

* Desktop with minimum 2GB RAM
* Internet connection (for Maven)

# Software Requirements

* Windows / Linux OS
* Eclipse IDE 4.2+
* Java 7+
* Maven (preferable)

# Outline

## Day 1

1. **Introduction**
   1. Concurrent vs Parallel Programming
   2. Throughput vs Latency
   3. Benefits of Threads
   4. Risks of Threads
2. **Concurrency Fundamentals**
   1. Thread Safety - Atomicity and Locking
   2. Sharing Objects - Visibility, Publication, Thread confinement and Immutability
   3. Composing Objects - Designing a thread-safe class and applying synchronization policies
   4. Building Blocks - Synchronized containers, Concurrent containers, Blocking queues, Synchronizers and Scalable Result Cache

## Day 2

1. **Structuring Concurrent Applications**
   1. Task Execution
   2. Executor Framework
   3. Task Cancellation and Shutdown
   4. Applying Thread Pools
   5. Fork/Join
2. **Liveness, Performance and Testing**
   1. Avoiding Liveness Hazards
   2. Deadlock - How to prevent, detect and correct
   3. Starvation
   4. Performance Impact and Scalability
   5. Testing Concurrent Programs

## Day 3

1. **Advanced Concurrency Topics**
   1. Explicit Locks
   2. Lock and Reentrant Lock
   3. Fairness
   4. Synchronized vs Reentrant Lock
   5. Read-write Locks
   6. Building Custom Synchronizers
   7. Atomic Variables and Non-blocking Synchronization
2. **Performance Considerations and Best Practices**