SE005J: ***Programming Techniques for High Performance Java Codes***

DURATION: 2 Days; Instructor-led

WHAT YOU WILL LEARN

This course introduces practical ways you can improve Java performance and presents more than 30 Java performance tuning and Java efficiency techniques. These techniques focus on Java language and library features. Performance is defined to include both speed and space issues, that is, how to make your programs run faster, while using less memory and disk space. Many of the techniques are illustrated by code examples.

The course raises a variety of performance issues and gives some hard numbers about how specific performance improvements work out. The ultimate aim of the course is to promote awareness of Java performance issues, so that you can write appropriate code in java.

High performance java application development is a large area. The course does not address all the challenges in dealing with Java related development. Instead, it focuses on to just the programming related aspects of java development. Other challenges are covered in more advanced courses.

Upon completion of this program, participants should be able to:

- Appreciate why performance is important issue in Java development.
- Understand the limitation of Java concerning the performance.
- Know some basic techniques in writing better performance java code.

AUDIENCE

This course is specially design for Java Programmers are concern about how to write efficient code.

PREREQUISITES:

- At least 6 months' experience in Java Programming.
- Know how to use Java application development tools.

METHODOLOGY

This program will be conducted with interactive lectures, PowerPoint presentations, discussions and practical exercises

COURSE OUTLINES

Module 1 - Introduction

- When to Worry About Performance?
- Performance Issues Not Covered in This Course
- Just-in-Time Compilers and Java Virtual Machines
- Environment and Tools Used in Code Examples
- Common heuristics in dealing with improving performance
- Understanding the benchmarks
- How Examples Were Timed?

Module 2 – Java Language Features

- Classes
- Default Constructors
- Constructor Chaining
- Class and Instance Initialization
- How the GC works?
- Recycling Class Instances
- Methods Inlining
- Using final keyword

Module 3 – String Class

- Strings
- Strings are Immutable
- Accumulating Strings Using char[] Arrays
- Using == and String.equals() to Compare Strings
- Interning Strings
- Obtaining the Length of a String
- Using toCharArray()
- Converting Strings to Numbers

Module 4 – Input and Output

- Buffering
- BufferedReader
- Formatting
- Obtaining File Information

Module 5 - Libraries

- System.arraycopy()
- Vector vs. ArrayList
- Setting Initial Array Capacity
- ArrayList vs. LinkedList

Module 6 – Performance by Language Features

- Bitwise Operators
- Use abstracts method instead of polymorphism
- Asynchronous Processing

Module 7 – Performance by Techniques

- Lookup Table Techniques
- Flyweight and Prototypes Design Patterns
- Avoid using iterators
- Use better Abstraction