Java 5 Update

Duration:	1 day
Type:	intermediate

Description

This workshop enables experienced Java developers to update their programming skills by learning and applying features introduced in the latest version of the language

Prerequisites

Delegates should be experienced Java developers

List of Modules

Miscellaneous Improvements

- Importing static members of a class
- Methods with variable argument lists
- Boxing primitive types into objects

Enhanced For Loop

- The syntax of the new loop
- Iterating over arrays and collections
- Iterating over generic collections
- Iterating over custom types via *Iterable*

Enumerated Types

- Why Java historically lacked enum support
- The typesafe enumeration design pattern
- Declaring enumerations in Java 5
- The link between enums and classes
- The *java.lang.Enum* base class
- Extending enums with new members
- Static methods added to enum types
- New collections which support enums

Annotations

- Adding metadata to Java code
- The advantages of annotations
- Annotations vs. configuration files
- Declaring Java 1.5 annotation types
- Understanding meta-annotations
- Adding methods to annotation types
- Defining default values for methods
- Discovering annotations with reflection

Generics in Java Part 1

- The evolution of Generics in Java
- Generics as a compile time construct
- Generics verses templates in C++
- Comparing Generics and inheritance
- Working with generic collection classes
- Introducing the *java.lang.Class<T>* type
- Support for Generics in the Reflection API

Generics in Java Part 2

- Declaring generic classes and methods
- Using the wildcard type in utility methods
- Defining constraints with bounded wildcards
- Adding generics to existing Java code

Threading Library Enhancements

- Changes made to the Java Memory Model
- Creating threads using *Executor* objects and futures
- More complex ways of acquiring and releasing locks
- New thread-friendly additions to the Collections library

Input/Output Enhancements

- String parsing with the *Scanner* class
- Pretty-printing with the *Formatter* class