

Course 471:
Java 5 and 6: New and Useful Features
(2 days)

This in-depth workshop is a best-practices course that provides not just an overview of features that are new in Java 1.4, Java 5 and Java 6, but also when and why to use these features. Students will learn how (and why) to write generic classes and methods, learn best practices on the use of wildcards and enums, provide enhanced-for iterability to their collections, define and process annotations, take advantage of JVM enhancements, manage beans remotely with JMX and enable end-user scripting. The course provides hands-on experience with the new concurrency framework and NIO.

Learning Objectives...

- Follow best practices in using new Java features such as generics, enums, annotations and scripting
- Improve performance by incorporating new concurrency and file handling techniques.

Who can benefit?

This is an intermediate level Java programming course, designed for developers who wish to learn what's new in Java. The student should be an experienced Java programmer, with practical development experience in Java 1.3.

See next page for a detailed course outline...



Course Outline

Chapter 1: Generics

- Avoiding runtime errors with generics
- Using generic collections classes
- Interoperation with non-generic code
- Checked collection wrappers
- Exercise 1: Using generic collections
- The pitfalls of erasure
- Casts, getClass() and instanceof
- Writing a generic class
- Exercise 2: Writing a generic service
- Writing a generic method
- Inferring return types from parameters
- Bounded Wildcards (super keyword)
- Wildcard capture
- Exercise 3: Writing a generic method with bounded wildcards

Chapter 2: Readability improvements

- Enhanced for-loop
- Exercise 4: Using Iterators
- Exercise 5: Using enhanced-for
- Iterable
- Exercise 6: Creating an Iterable collection
- Limitations of the enhanced for-loop
- Autoboxing
- Autoboxing caveats
- Variable arguments
- Limiting range and enforcing type
- The Java enum pattern
- Lists of constants
- Enums with constructors
- Performance improvements with enum
- Enum serialization
- Constant interface pattern
- Exercise 7: Writing an enum
- Static imports

Chapter 3: Annotations

- Metadata
- Annotations in java.lang
- Need for @Override
- Writing an annotation
- Retention policies
- Processing an annotation
- Exercise 8: Defining and processing annotations

Chapter 4 [Optional]: JVM Improvements

- Sharing class data
- Setting targets for garbage collection
- Shutdown hooks
- Chained exceptions
- Environment variables
- JVM Tool Interface



- Managing Beans remotely with JMX
- Exposing a bean for remote management
- Monitoring a JVM with JConsole

Chapter 5: Useful core classes

- Scanner and Formatter
- StrictMath vs. Math
- Timer and TimerTask
- Exercise 9: Scheduling timer tasks and formatting output
- Singleton and empty collections
- Deque
- Regular expressions
- Preferences API
- Exercise 10: Working with the preferences API
- Logging API
- Log levels, Loggers, Handlers
- Formatters, Filters
- Configuration file

Chapter 6: Concurrency Framework

- Threads
- Synchronization
- Monitors
- Concurrent collections
- Atomic variables
- Executors framework
- Exercise 11: Using the executors framework
- Semaphore
- CountdownLatch
- CyclicBarrier
- Exchanger
- ReadWriteLock

Chapter 7: NIO

- Writing scalable servers
- Buffers of primitives
- Reading and writing files
- Exercise 12: File I/O with NIO
- Locking files
- Memory mapping
- Non-blocking sockets
- Channel transfers

Chapter 8 [Optional]: Scripting

- Using the scripting API
- Evaluating a script
- Applying security to scripted classes

Please contact your ROI representative to discuss course customization!!!