

Intermediate Java Programming (Java 110)

Length:

4 days

Description:

This course goes in depth in many of the topics covered in the Java Fundamentals class. It also introduces JDBC, Networking, Systems Programming, JUnit, Log4j and ANT.

Course Objectives:

- Provides an increased understanding of the Java programming language
- Reinforces the lectures with extensive hands-on programming exercises

Audience:

Java programmers and system architects who want an in-depth working knowledge of the Java language, its APIs, and its idiosyncrasies

Prerequisites:

- A working knowledge of the Java language.
- Completion of a class in Java Fundamentals and Object Oriented Programming.
- A strong understanding of Object-Oriented Analysis and Design.

Course Content:**Chapter 1 - Collections and Arrays**

- Objectives
- One Name, Many Items
- Collections are Object Containers
- The Old Favorites
- Using Vectors
- Hashtables
- Other Old-Time Favorites
- New and Improved Collections
- Collection Hierarchy
- The Collection Interface
- The Iterator
- The List Interface
- The ArrayList Class
- The LinkedList Class
- The Map Interface
- The HashMap Class
- The SortedMap Interface
- The Comparable Interface
- Comparable & Comparator Examples
- The TreeMap Class
- Set and SortedSet
- Collection Classes: Old vs. New
- Final Notes

Chapter 2 - I/O Streams

- Overview
- Types and Uses
- The File Class
- Byte Streams

- Data Streams
- Conversion Streams
- Readers and Writers
- Text Streams
- Object Streams
- Other Streams
- Final Notes

Chapter 3 - JDBC

- Overview
- Drivers
- Basic Steps
- ResultSets
- MetaData
- Statements
- Prepared Statements
- Exceptions
- Transactions
- Final Notes
- **Threads**
- Overview
- Threaded Apps
- Thread States
- Synchronization
- Runnable Interface
- Interrupting Threads
- Thread Cooperation
- Collections Revisited
- Final Notes

Chapter 4 - Networking

- Overview
- Protocols
- Types of Transfer
- The TCP/IP Protocol Suite
- TCP Streams
- Sockets
- ServerSockets
- MultiThreaded Servers
- Datagrams
- Final Notes

Chapter 5 - Systems Programming

- The System Class
- Creating Processes
- Shutdown
- More on Run Time
- Security
- Standard Packages

Chapter 6 - Unit Testing and JUnit

- Objectives
- Why Unit Test?
- Why Don't Developers Unit Test?
- Successful Unit Testing
- Benefits of Automated Testing
- Unit Testing Best Practices
- Overview of JUnit
- What is JUnit?
- Steps for Testing in JUnit
- JUnit Assert Methods
- JUnit Fail Methods
- A Business Object to Test
- Create Project for Test Case
- Add JUnit to the Build Path
- Create the Test Case
- Test Case Parts
- Run a Test Case
- What Should You Test?
- How Should You Test?
- Test Case Failure
- Test Fixtures
- Testing Exceptions
- Test Suites
- Best Practices
- Review

Chapter 7 - Log4j

- Objectives
- Logging
- Log4j
- Overview
- Terminology
- When to Use It
- Best Practices
- J2EE Considerations
- Documentation and Information
- Review

Chapter 8 - ANT Overview

- Objectives
- Overview
- Working with Java
- Core Ant Concepts
- Invoking Ant
- Build Files
- Projects
- Projects in XML
- Targets
- Targets in XML
- Targets which depend on Other Targets
- Example Dependencies
- Example Dependencies in XML

- Tasks
- Properties
- Filesets
- Working with Paths
- Java Applications in JARs, Running JUnit
- Simple Complete Ant Build File
- Javac Task
- Jar Task
- JUnit Task
- ANT 1.6
- Application Architecture Common Build Files
- Common SWA Build Example
- Integration with WSAD
- Support and Additional Information
- Review