



## Course Content

### Course Description:

Client/server applications can be written utilizing both the java.net and java.rmi packages. Security is also covered in detail throughout several chapters including security policies and data encryption. XML parsing is done using the JAXP API. Programmers can use C and C++ functions from their Java programs using JNI. The course ends with several chapters on design patterns and how they can be put to best use in different types of Java applications.

### Prerequisites:

Java Programming course or equivalent knowledge.

### Topics:

#### Advanced I/O- Object Serialization

- What is a Serialization?
- Serializable Objects
- Writing an Object
- Reading an Object
- Handling Exceptions
- Customizing Serialization
- Controlling Serialization
- Versioning

#### Advanced I/O-New I/O

- The java.nio package
- Buffers and Channels
- Buffer Implementations
- Buffer Methods
- ByteBuffer Methods
- FileChannel
- File Locking
- MappedByteBuffer
- Transferring Data Between Channels
- Character Sets

#### Reflection

- Introduction to Reflection
- The Class Class
- The Reflect Package
- Constructors
- Fields

- Methods
- Exception Handling and Reflection
- JavaBeans
- Dynamic Programming

#### Advanced JDBC

- JDBC SQL Escape Syntax
- The Execute() Method
- Batch Updates
- Updatable Result Sets
- Large Objects
- Working with Savepoints
- RowSets
- CachedRowSets
- DataSources

#### Networking with Sockets

- Clients and Servers
- Ports, Addresses and Protocols
- The Socket Class
- Communication Using I/O
- Servers
- The Server Socket Class
- Concurrent Servers
- The URL Class
- The URL Connection Class



# Advanced Java Programming

Course ID#: 1260-150-ZZ-W

35 Hrs

## Remote Method Invocation

- Distributed Applications
- Stubs
- Steps to Create a Remote Object
- An RMI Client
- An RMI Server
- RMI Classes and Interfaces
- Class Distribution
- RMI Utilities
- Parameter Passing and Serialization

## Advanced RMI

- Client Callbacks
- Dynamic Class Loading
- Activation
- Activatable Objects
- Registering Activatable Objects
- Security and Activation
- JNDI and RMI Registry
- RMI-IIOP

## Managing Security Policies

- Untrusted Code
- Security Managers
- The Java Security Model
- Policy Entries
- Policy Files
- Using the Policy Tool
- Securing Applets
- Securing Applications

## Keys, Signatures, and Certificates

- Jar Files
- Data Security Concerns
- Message Digests
- Digital Signatures
- Using Keytool
- Using jarsigner
- Certificates
- Certificate Chains
- Managing Keys and Certificates
- Security Policies for Signed Code
- Java Cryptography Architecture

## Encryption with the Javax.Crypto Package

- Cryptography Concepts
- Encryption Keys
- Cipher Algorithms
- Modes and Padding Schemes
- The Cipher Class
- Encrypting and Decrypting Data
- Cipher Input Stream
- Encryption Using Password Ciphers
- Exchanging Encrypted Keys
- Sealed Objects

## Java Authentication and Authorization Service

- (JAAS)
- Authentication and Authorization
- JAAS Overview
- LoginContext
- Subjects, Principles and PrivilegedActions
- Authentication and the NTLoginModule
- Defining Permissions in Policy Files
- KeyStoreLoginModule
- Callbacks
- NameCallback and PasswordCallback
- The Policy Class

## Java Naming and Directory Interface (JNDI)

- Naming and Directory Services
- Namespaces and Contexts
- Naming Operations
- Bindings
- Attributes
- Directory Observations
- DNS Lookups with JNDI
- JNDI in J2EE



# Advanced Java Programming

Course ID#: 1260-150-ZZ-W

35 Hrs

## Parsing XML with Java-JAXP

- The Java API for XML Processing
- Introduction to SAX Parsing
- SAX Parser and JAXP
- Sax Event Methods
- Introduction to DOM
- Parsing DOM with JAXP
- The DOM API
- Validation
- Transformation

## Native Methods

- Overview of Java Native Methods and JNI
- How to Create and Use Native Methods
- Native Method Declaration
- Using javah
- Creating the Implementation Code
- Compilation
- Distribution
- Using the Native Methods
- JNI
- Passing Arguments
- Calling Java Methods in Native Code
- JNI Signatures

## Java Design Patterns—Creational Patterns

- What are Design Patterns?
- What are Creational Patterns?
- Singleton--Introduction
- Singleton-Implementation
- Singleton—When to Use?
- Factory Method--Introduction
- Factory Method--Implementation
- Factory Method—When to Use?
- Builder--Introduction
- Builder--Implementation
- Builder—When to Use?

## Java Design Patterns—Structural Patterns

- What are Structural Patterns?
- Façade--Introduction
- Façade--Implementation
- Façade—When to Use?
- Adapter--Introduction

- Adapter--Implementation
- Adapter—When to Use?
- Composite--Introduction
- Composite--Implementation
- Composite—When to Use?

## Java Design Patterns—Behavioral Patterns

- What are Behavioral Patterns?
- Template-Introduction
- Template--Implementation
- Template—When to Use?
- State--Introduction
- State--Implementation
- State—When to Use?
- Observer--Introduction
- Observer--Implementation
- Observer—When to Use?

## Appendix A—JDBC SQL Programming

- Error Checking and the SQLException Class
- The SQL Warning Class
- JDBC Types
- Executing SQL Queries
- ResultSetMetaData
- Using a Prepared Statement
- Parameterized Statements
- Stored Procedures
- Transaction Management

## Appendix B--Eclipse

- Introduction to Eclipse
- Installing Eclipse
- Running Eclipse for the First Time
- Editors, Views, and Perspectives
- Setting Up a Project
- Creating a New Java Application
- Running a Java Application
- Debugging a Java Application
- Shortcut Key Sequences
- Setting the Classpath
- Importing Existing Java Code into Eclipse