

**Course Description:** Advanced Java is a comprehensive study of many advanced Java topics. These include assertions, collection classes, searching and sorting, regular expressions, logging, bit manipulation, serialization, threads, networking with sockets, Remote Method Invocation, and Java Database Connectivity.

**Who Should Attend:** This course is intended for programmers who have been programming in Java and who wish to write programs in Java using many of the advanced Java features.

**Prerequisites:** Students should have completed a beginning Java course or have programmed in Java for at least three - six months.

**Benefits of Attendance:** Upon completion of this course, students will be able to:

- Document and package a Java application
- Use many of the new enhancements added to the Java API
- Use assertions to write robust Java code.
- Use regular expressions for efficient pattern matching.
- Choose appropriate data structures from the Java Collection API.
- Sort and search arrays and lists using a variety of techniques.
- Capture configuration and debugging information using the Java Logging APIs.
- Use Generics to create type safe collections.
- Serialize Java objects.
- Use features of the New I/O API.
- Write TCP/IP Client Server applications using sockets.
- Write multi-threaded Java applications.

### Course Outline:

#### Review of Java Fundamentals

The Java Environment  
Data Types  
Strings  
StringBuffers  
Arrays  
Passing Data Types to a Method  
Constructors and Initialization  
Inheritance  
Abstract Classes  
Interfaces  
Static Data, Methods and Blocks  
Wrapper Classes

#### Packaging a Java Application

Introduction  
Packages  
Managing Source and Class Files  
jar Files  
The Manifest File  
The javadoc Utility  
Documenting Classes and Interfaces  
Documenting Fields  
Documenting Constructors and Methods

#### Miscellaneous Enhancements

Enhanced for Loop  
Autoboxing and Auto-Unboxing  
Static Imports  
Varargs  
Typesafe Enums  
Formatted Strings  
Format Specifier Syntax  
Format Specifier Conversions  
Format Specifier Flags  
Formatted Integers Example  
Formatted Floating Points Example  
Formatted Strings Example  
Formatted Dates Example  
Complex Formatted Example

#### Assertions

Introduction  
Assertion Syntax  
Compiling with Assertions  
Enabling and Disabling Assertions  
Assertion Usage

#### Regular Expressions

Regular Expressions  
String Literals  
Character Classes  
Quantifiers  
Capturing Groups and Backreferences  
Boundary Matchers

#### Pattern and Matcher

#### The Java Collection Classes

Introduction  
The Arrays Class  
Searching and Sorting Arrays of Primitives  
Sorting Arrays of Objects  
The Comparable and Comparator Interfaces  
Sorting - Using Comparable  
Sorting - Using Comparator  
Collections  
Lists and Sets  
Iterators  
Lists and Iterators Example  
Maps  
Maps and Iterators Example  
The Collections Class  
Rules of Thumb

#### Generics

Introduction  
Defining Simple Generics  
Generics and Subtyping  
Wildcards  
Bounded Wildcards  
Generic Methods

#### Advanced I/O

Introduction  
Basic File I/O Example  
Buffered I/O  
The Console Class  
Object Serialization  
Serialization Issues  
Compressed Files  
Zip File Example  
Writing Your Own I/O Classes  
Property Files  
The Preferences Class  
Exercises

#### Enhanced I/O

Introduction  
Channels  
Buffers  
Typed Buffers  
Direct Buffers

#### Logging API

Introduction  
Loggers  
Logger Levels  
Logger Handlers  
Specifying Handlers and Formatters  
Configuring Handlers

#### LogManager

#### Networking

Networking Fundamentals  
The Client/Server Model  
InetAddress  
URLs  
Sockets  
A Time-of-Day Client  
Writing Servers  
Client/Server Example

#### Threads and Concurrency

Review of Fundamentals  
Creating Threads by Extending Thread  
Creating Threads by Implementing Runnable  
Advantages of Using Threads  
Daemon Threads  
Thread States  
Thread Problems  
Synchronization  
Performance Issues  
Exercises

#### Remote Method Invocation (RMI)

Introduction  
RMI Architecture  
The Remote Interface  
The Remote Object  
Writing the Server  
The RMI Compiler  
Writing the Client  
Remote Method Arguments and Return Values  
Dynamic Loading of Stub Classes  
Remote RMI Client Example  
Running the Remote RMI Client Example  
Exercises

#### Java Database Connectivity (JDBC)

Introduction  
Relational Databases  
Structured Query Language  
A Sample Program  
Transactions  
Meta Data  
Exercises