



ADVANCE JAVA

Course Overview

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.

COURSE CONTENTS:

- **REVIEW OF JAVA FUNDAMENTALS**
- **PACKAGING AND DISTRIBUTING A JAVA APPLICATION**
- **MISCELLANEOUS ENHANCEMENTS**
- **ASSERTIONS**
- **REGULAR EXPRESSIONS**
- **THE JAVA COLLECTION CLASSES**
- **GENERICS**
- **ADVANCED I/O & ENHANCED I/O**
- **LOGGING API, NETWORKING**
- **THREADS AND CONCURRENCY**
- **REMOTE METHOD INVOCATION (RMI)**
- **JAVA DATABASE CONNECTIVITY (JDBC)**

COURSE CONTENT DETAILS:

- **REVIEW OF JAVA FUNDAMENTALS**
 1. The Java Environment
 2. Data Types
 3. The String Class

4. The String Buffer Class
 5. Arrays
 6. Passing Data Types to a Method
 7. Constructors and Initialization
 8. Inheritance
 9. Abstract Classes
 10. Interfaces
 11. Static Data, Methods, and Blocks
 12. Wrapper Classes
 13. I/O
- **PACKAGING AND DISTRIBUTING A JAVA APPLICATION**
 1. Packages
 2. Managing Source and Class Files
 3. The javadoc Utility
 4. Documenting Classes and Interfaces
 5. Documenting Fields

6. Documenting Constructors and Methods
7. Running the `java doc` Utility
8. `jar` Files
9. The Manifest File
10. Bundling and Using Jar-Packaged Resources
- **MISCELLANEOUS ENHANCEMENTS**
 1. Enhanced `for` Loop
 2. Autoboxing and Auto-Unboxing
 3. Static Imports
 4. `Var Args`
 5. Typesafe Enums
 6. Formatted Strings
 7. Format Specifier Syntax
 8. Format Specifier Conversions
 9. Format Specifier Flags
 10. Formatted Integers Example
 11. Formatted Floating Points Example
 12. Formatted Strings Example
 13. Formatted Dates Example
 14. Complex Formatted Example
- **ASSERTIONS**
 1. Introduction
 2. Assertion Syntax
 3. Compiling with Assertions
 4. Enabling and Disabling Assertions
 5. Assertion Usage
- **REGULAR EXPRESSIONS**
 1. Regular Expressions
 2. String Literals
 3. Character Classes
 4. Quantifiers
 5. Capturing Groups and Back references
6. Boundary Matchers
7. Pattern and Matcher
- **THE JAVA COLLECTION CLASSES**
 1. Introduction
 2. The Arrays Class
 3. Searching and Sorting Arrays of Primitives
 4. Sorting Arrays of Objects
 5. The Comparable and Comparator Interfaces
 6. Sorting - Using Comparable
 7. Sorting - Using Comparator
 8. Collections
 9. Lists and Sets
 10. Iterators
 11. Lists and Iterators Example
 12. Maps
 13. Maps and Iterators Example
 14. The Collections Class
 15. Rules of Thumb
- **GENERICS**
 1. Introduction
 2. Defining Simple Generics
 3. Generics and Subtyping
 4. Wildcards
 5. Bounded Wildcards
 6. Generic Methods
- **ADVANCED I/O**
 1. Introduction
 2. Basic File I/O Example
 3. Buffered I/O
 4. The `Console` Class
 5. Object Serialization
 6. Serialization Issues
 7. Compressed Files
 8. Zip File Example
 9. Writing Your Own I/O Classes
 10. Property Files

11. The Preferences Class

- **ENHANCED I/O**

1. Introduction
2. Channels
3. Buffers
4. Typed Buffers
5. Direct Buffers

- **LOGGING API**

1. Introduction
2. Loggers
3. Logger Levels
4. Logger Handlers
5. Specifying Handlers and Formatters
6. Configuring Handlers
7. LogManager

- **NETWORKING**

1. Networking Fundamentals
2. The Client/Server Model
3. InetAddress
4. URLs
5. Sockets
6. A Time-of-Day Client
7. Writing Servers
8. Client/Server Example

- **THREADS AND CONCURRENCY**

1. Review of Fundamentals
2. Creating Threads by Extending Thread
3. Creating Threads by Implementing Runnable
4. Advantages of Using Threads
5. Daemon Threads
6. Thread States
7. Thread Problems
8. Synchronization
9. Performance Issues

- **REMOTE METHOD INVOCATION (RMI)**

1. Introduction
2. RMI Architecture
3. The Remote Interface
4. The Remote Object
5. Writing the Server
6. The RMI Compiler
7. Writing the Client
8. Remote Method Arguments and Return Values
9. Dynamic Loading of Stub Classes
10. Remote RMI Client Example
11. Running the Remote RMI Client Example

- **JAVA DATABASE CONNECTIVITY (JDBC)**

1. Introduction
2. Relational Databases
3. Structured Query Language
4. A Sample Program
5. Transactions
6. Meta Data