

Course Catalog
2012-2013

ADVANCED Java Training



Advanced Java Programming

CSLiT Training's Advanced Java™ Programming training course teaches Java developers a set of advanced Java development skills, including generics, threads, reflection, annotations, and sockets.

Course Objectives

Advanced Java Training Objectives:

- Understand Java as a purely object-oriented language, and implement software as systems of classes.
- Implement and use inheritance and polymorphism, including interfaces and abstract classes.
- Design appropriate exception handling into Java methods.
- Use the standard logging API to write diagnostic information at runtime.
- Understand the structure of streams in Java, and learn how to use streams to manage file I/O.

Learn how to use Java Serialization to internalize and externalize potentially complex graphs of objects.

Who Should Attend

Java programmers who wish to increase their depth of knowledge in Java programming and explore the uses of the various advanced packages.

Prerequisite

Students should have solid Java programming experience, especially object-oriented use of the language. Language features and techniques that are integral to some lab exercises include interfaces and abstract classes, threading, generics and collections, and recursive methods. Beginning Java Application Development class is recommended.

Course Duration

48 Hours, 16 Classes, 3 Hours per class

Course Details

Lesson 01: Review of Java Fundamentals

- The Java Architecture
- Forms for Java Software
- Three Platforms
- The Java Language
- Numeric Types
- Characters and Booleans
- Enumerations
- Object References
- Strings and Arrays
- Conditional Constructs
- Looping Constructs
- Varargs

Lesson 02: Object-Oriented Software

- Complex Systems
- Abstraction
- Classes and Objects
- Responsibilities and Collaborators
- UML
- Relationships
- Visibility

Lesson 03: Classes and Objects

- Java Classes
- Constructors and Garbage Collection
- Naming Conventions and JavaBeans
- Relationships Between Classes
- Using this
- Visibility
- Packages and Imports
- Overloading Methods and Constructors
- JARs

Lesson 04: Inheritance and Polymorphism in Java

- UML Specialization
- Extending Classes
- Using Derived Classes
- Type Identification

Course Details

Lesson 05: Using Classes Effectively

- Class Loading
- Static Members
- Statics and Non-Statics
- Static Initializers
- Static Imports
- Prohibiting Inheritance
- Costs of Object Creation
- Strings and String Buffers
- Controlling Object Creation
- Understanding Enumerated Types
- Stateful and Behavioral Enumerations

Lesson 06: Interfaces and Abstract Classes

- Separating Interface and Implementation
- UML Interfaces and Realization
- Defining Interfaces
- Implementing and Extending Interfaces
- Abstract Classes

Lesson 08: Collections

- Dynamic Collections vs. Arrays
- UML Parameterized Type
- Generics
- Using Generics
- The Collections API
- The Collection<E> and List<E> Interfaces
- The Array List<E> and Linked List<E> Classes
- Looping Over Collections: Iterable <E>
- Collecting Primitive Values: Auto-Boxing
- Using Wildcards with Generic Types
- Iterators and the Iterator <E> Interface
- Maps and the Map<K,V> Interface
- Sorted Collections
- The Sorted Set<E> and Sorted Map<K,V> Interfaces
- The Collections Class Utility
- Algorithms
- Conversion Utilities

Course Details

Lesson 14: Java Serialization

- The Challenge of Object Serialization
- Serialization API
- Serializable Interface
- Object Input Stream and Object Output Stream
- The Serialization Engine
- Transient Fields
- Read Object and write Object
- Externalizable Interface

Lesson 15: Automated Unit Testing with JUnit

- Automated Testing
- JUnit and Related Tools
- The @Test Annotation
- The Assert Class Utility
- Test Runners
- Lifecycle Methods

Lesson 16: Generics

- Using Generics
- Type Erasure
- Type Boundaries
- Wildcards
- Generic Methods
- Strengths and Weaknesses of Generics
- Legacy Code and Generics

Lesson 17: Threads

- Java Thread Model
- Creating and Running Threads
- Manipulating Thread State
- Thread Synchronization
- Volatile Fields vs. Synchronized Methods
- wait and notify
- join and sleep
- The Concurrency API
- Atomic Operations
- Thread Pools

Course Details

Lesson 14: Java Serialization

- The Challenge of Object Serialization
- Serialization API
- Serializable Interface
- Object Input Stream and Object Output Stream
- The Serialization Engine
- Transient Fields
- Read Object and write Object
- Externalizable Interface

Lesson 15: Automated Unit Testing with JUnit

- Automated Testing
- JUnit and Related Tools
- The @Test Annotation
- The Assert Class Utility
- Test Runners
- Lifecycle Methods

Lesson 16: Generics

- Using Generics
- Type Erasure
- Type Boundaries
- Wildcards
- Generic Methods
- Strengths and Weaknesses of Generics
- Legacy Code and Generics

Lesson 17: Threads

- Java Thread Model
- Creating and Running Threads
- Manipulating Thread State
- Thread Synchronization
- Volatile Fields vs. Synchronized Methods
- wait and notify
- join and sleep
- The Concurrency API
- Atomic Operations
- Thread Pools

Course Details

Lesson 18: Reflection

- Uses for Meta-Data
- The Reflection API
- The Class<T> Class
- The java.lang.reflect Package
- Reading Type Information
- Navigating Inheritance Trees
- Dynamic Instantiation
- Dynamic Invocation
- Reflecting on Generics

Lesson 19: Annotations

- Aspect-Oriented Programming and Java
- The Annotations Model
- Annotation Types and Annotations
- Built-In Annotations
- Annotations vs. Descriptors (XML)

Lesson 20: Sockets

- The OSI Reference Model
- Network Protocols
- The Socket Class
- The Server Socket Class
- Connecting Through URL Objects
- HTTP and Other TCP Servers
- Datagram Clients and Servers
- Non-Blocking Sockets
- Multi-Threading in Servers

