



Why is this course different from other JAVA courses

Traditional technical courses in computer languages usually cover only the technical aspects of the language, without their application areas or provide associated field level training. In today's competitive market, just having technical knowledge is not enough - the industry demands you to have considerable understanding of its application domain. For example, technical requirements in the financial industry usually require you to have knowledge about associated fields - products related to equity, derivatives, fixed income, mutual funds or offerings like prime brokerage, stock loaning or locates.

This course aims to provide you not only with core java or j2EE concepts but also empower you with the financial concepts and practices which are most popular and sought after in today's financial industry.

Whats in it for you?

This course will provide you with the basic and advanced j2EE concepts that is focussed towards getting you a job or providing you with an opportunity to move to a better position in the industry. Whether you are a person looking to get a suitable job or switch to a more high paying financial industry opportunity - this course is for you. The training is structured so that you not only know the technologies required but also know the financial products onto which the technology is applied.

We have an evaluation based training approach - We evaluate you at regular and strategic intervals. This not only lets you know where you stand but also provides an opportunity to revisit modules where you feel you lack confidence.

We at CLD have extensive marketing potential and vast experience in fitting the right profile to the right requirements for full time or consultant roles in the financial industry. This course also gives you the chance to get noticed in the class, so that it can interest us in you to offer a position or a great job in the industry.

Who is the instructor and where are the classes?

Instructor:

Souvik Sarkar

Location of the classes

CLD Associates,
190 Lincoln Highway,
Edison, NJ 08820

Training objectives

- To teach attendees the fundamentals of Java programming and how to use Java to write applications
- To teach attendees core object-oriented concepts, including classes, packages, objects, methods, properties, abstraction, polymorphism, inheritance, encapsulation, and more
-
- To impart the concepts and working of different financial products like equities, derivatives, Credit derivatives, Collateralized debt obligations, forex trading and mutual funds
- To provide an understanding of different financial product and functions like prime brokerage, trading, etc.

Training prerequisites

This Java training course can be tailored for audiences ranging from first-time programmers to experienced OO developers seeking to learn Java. You are not required to have any prior java experience but having a basic knowledge of programming would help. You are also not required to have any prior financial domain knowledge.

Course Details - part 1

- **Getting Started with Java SE**
 - What is Java and IDE?
 - How to Get Java and how to set up a programming environment
 - A First Java Program
 - Compiling and Interpreting languages and how does it work

- **Packages**
 - Packages
 - The import Statement
 - Static Imports
 - CLASS PATH and Import
 - Defining Packages
 - Package Scope

- **Datatypes and Variables**
 - Primitive Datatypes
 - Declarations
 - Variable Names
 - Numeric Literals
 - Character Literals
 - String
 - String Literals
 - Arrays
 - Non-Primitive Datatypes
 - The Dot Operator

- **Operators and Expressions**
 - Expressions
 - Assignment Operator
 - Arithmetic Operators
 - Relational Operators
 - Logical Operators
 - Increment and Decrement Operators
 - Operate-Assign Operators (+=, etc.)
 - The Conditional Operator
 - Operator Precedence
 - Implicit Type Conversions
 - The Cast Operator

- **Control Flow**
 - Statements
 - Conditional (if) Statements
 - Adding an else if
 - Conditional (switch) Statements
 - while and do-while Loops
 - for Loops
 - A for Loop Diagram
 - Enhanced for Loop
 - The continue Statement
 - The break Statement

- **Methods**
 - Methods
 - Calling Methods
 - Defining Methods
 - Method Parameters
 - Scope
 - So, Why All the static?

- **Object-Oriented Programming**
 - Introduction to Object-Oriented Programming
 - Classes and Objects
 - Fields and Methods
 - Encapsulation
 - Access Control
 - Inheritance
 - Polymorphism
 - Best Practices

- **Objects and Classes**
 - Defining a Class
 - Creating an Object
 - Instance Data and Class Data
 - Methods
 - Constructors
 - Access Modifiers
 - Encapsulation

- **Using Java Objects**
 - Printing to the Console
 - printf Format Strings
 - Popular java objects String, StringBuilder and StringBuffer
 - Methods and Messages
 - Mutable and immutable objects
 - Parameter Passing
 - Comparing and Identifying Objects

- Destroying Objects
- Using the Primitive-Type Wrapper Classes
- Auto boxing and un-boxing

- **EVALUATION 1**

- **Inheritance in Java**
 - Inheritance
 - Inheritance in Java
 - Casting
 - Method Overriding
 - Polymorphism
 - super
 - The Object Class

- **Advanced Inheritance and Language Constructs**
 - Abstract Classes
 - Interfaces
 - Using Interfaces
 - Using abstract classes

- **Exception Handling**
 - Exceptions Overview
 - Catching Exceptions
 - The finally Block
 - Exception Methods
 - Declaring Exceptions
 - Defining and Throwing Exceptions
 - Errors and RuntimeExceptions
 - Assertions
 - final, finally and finalize

- **Input/Output Streams**
 - Overview of Streams
 - Bytes vs. Characters
 - Converting Byte Streams to Character Streams
 - File Object
 - Binary Input and Output
 - PrintWriter Class
 - Reading and Writing Objects
 - Basic and Filtered Streams

- **Core Collection Classes**
 - The Collections Framework
 - The Set Interface
 - Set Implementation Classes
 - The List Interface

- List Implementation Classes
- The Queue Interface
- Queue Implementation Classes
- The Map Interface
- Map Implementation Classes

- **Collection Sorting and Tuning**
 - Using Java 5.0 Features with Collections
 - Sorting with Comparable
 - Sorting with Comparator
 - Sorting Lists and Arrays
 - Collections Utility Methods
 - ArrayList
 - HashMap and HashSet

- **Introduction to JDBC**
 - The JDBC Connectivity Model
 - Database Programming
 - Connecting to the Database
 - Creating a SQL Query
 - Getting the Results
 - Updating Database Data

- **Introduction to Threads**
 - Non-Threaded Applications
 - Threaded Applications
 - Creating Threads
 - Thread States
 - Runnable Threads
 - Coordinating Threads
 - Interrupting Threads
 - Runnable Interface
 - ThreadGroups

- **EVALUATION 2**

Course Details - part 2

- **Introduction to financial applications**
 - Computers and finance
 - How is java used in finance industry

- **Financial products - Equity**
 - Trading concepts - OTC and exchange traded
 - settlement rules
 - what is Market making

- algorithmic trading
- **Financial concepts - Derivatives**
 - Credit derivatives
 - Contracts and settlements of derivative products
 - Interest rate products
 - Collateralized debt obligations - Tranches
- **Financial concepts - Fixed income and Foreign exchange**
 - Concepts of fixed income products
 - Bonds - structured and unstructured
 - Municipal bonds
 - FX trading and structure
- **Financial products and common client offerings**
 - Prime brokerage
 - Stock loans
 - E-locates
 - Custody
 - Portfolio management and advisory investment functions
 - Deal making and underwriting IPOs
- **EVALUATION 3**