

Learn Java (Training Course For Programmers)

Summary:

This intensive 5-day introductory course is aimed at programmers with experience in at least one other programming language.

Delegates will, over the course of the week, build a simple but fully functional sample application through attempting a series of graded exercises.

This training course covers a wide range of features in the Java language. Our aim is that delegates should finish the course having gained some fluency in writing code. This is not merely a theoretical course, and as such requires a high level of commitment from delegates who wish to make the most of the training.

All topics are backed up with a series of code examples which delegates will have the opportunity to run and modify in order to gain a deeper understanding of the Java language.

Our trainers are Java experts who have worked for a number of years as developers, architects and consultants in industry as well as being experienced in educating other professionals.

Objectives:

This course will enable delegates new to Java to write and understand Java code which follows appropriate object-oriented convention and relating to the following areas:

- Language Fundamentals – classes, objects, references, object lifecycle and the JVM.
- Operators and Program Control
- Object-Oriented features: inheritance, polymorphism, interfaces, constructors
- Core Features: exception handling, constructors, packages, inner classes
- Advanced Features: threads, generics, collections, RMI, JDBC, i/o, garbage collection, aliasing and Servlet/JSP web programming.

Topics Covered in Depth Include:

- Introduction
 - The JVM
 - Compilation and Execution Process
 - The main() method
 - Temporary Variables
 - The Eight Primitive Types
 - Simple Operations
 - Overflow
 - Casting between types
 - Program Control - if, else, while, switch
 - Basic and Enhanced For loop
 - break/continue
 - Increment/Decrement operators
 - Constants
- Classes
 - java.lang.String as an example
 - Methods
 - The main() method
 - Instance Variables
 - Temporary Variables
 - Object Instances
 - Constructors
 - this() and super()
 - Examining Javadoc
- Object Relationships
 - Object References
 - Object Composition
 - Aliasing
 - Static data
 - Packages, Imports
 - "this"
- Inheritance and Polymorphism
 - extending classes
 - Overriding vs Overloading methods
 - Order of Constructor Execution
 - Abstract Classes
 - Interfaces

- Polymorphism
- Solid Object-Oriented Principles
 - Encapsulation: public, private, protected
 - Principle of Least Knowledge
 - Favour Composition over Inheritance
 - Use interfaces where practical
 - Refactoring
 - Object-Oriented Patterns
 - final keyword
- Exception Handling
 - Runtime Exceptions
 - User Defined Exceptions
 - try, catch, finally
 - Understanding Exception Subtleties
- Miscellaneous Topics
 - Garbage Collection
 - toString() method
 - Arrays
 - Inner Classes
 - Anonymous Inner Classes
 - Wrapper Classes - Decorator Pattern
 - The equals() operator
 - Program arguments
 - Virtual Machine Arguments
 - Packaging your code in a JAR file
- Practical Considerations
 - Choice of IDE
 - Refactoring Tools
 - Unit Testing to Improve Code Quality
 - JUnit as the de facto Standard
 - setUp() and tearDown()
 - Refactoring
 - Maintainability & Testing through Proper O-O
- Threads
 - How threads work - single and multiple processors
 - Runnable interface
 - run() and start()
 - Inter-thread communication
 - Thread Safety
 - Synchronization to protect resources
- Deadlock
- Runnable, Running, Blocked and Waiting states
- Generics, Collections and Miscellaneous Topics
 - Autoboxing
 - Varargs
 - Java Generics
 - Parameterized Types
 - Storing and Manipulating multiple objects
 - Collection, Map, List, Set, Queue Interfaces
 - List implementations: Vector, ArrayList, LinkedList
 - equals() and hashCode()
 - Iterating through a Collection
 - Type Erasure
 - Sorting and Manipulating Collections
- RMI
 - Java objects communicating across a network
 - The RMI Registry
 - The RMI Compiler
 - Client, Server and Stub
- JDBC
 - Driver classes
 - Executing SQL through Statements
 - ResultSet - the data returned
 - Exception handling in JDBC
 - Alternatives to JDBC
- Java i/o
 - Streams, Writers, Readers, Console
 - Java i/o libraries
 - Decorating to add functionality
- Web Programming in Java
 - A simple servlet
 - A simple JSP
 - The web.xml configuration file
 - Deploying to an Application Server
 - web.xml configuration - making your code available on the web

Pre-Requisites:

This course is aimed at programmers and other technology specialists who have a working knowledge of at least one other programming language.

Contact:

Please contact us at training@codeclass.co.uk or phone us on 0121 314 4347 if you are interested in delivery of this course on your site and to discuss our extremely competitive rates.