

<p>LN 029 5 days</p>	<h2>Java™ Programming</h2> <p><b>Course Description:</b></p> <p>This intensive course provides a solid introduction to the Java programming language and development environment. In this fast-paced, hands-on course, programmers acquire the skills they need to write standalone applications in Java.</p> <p>The course is designed to teach the syntax of the Java programming language; object-oriented programming with the Java programming language; creating graphical user interfaces (GUI); exceptions; file input/output (I/O); threads and networking; database access with JDBC. We pay particular attention to testing and test development using the JUnit testing frame.</p>	
	<p><b>Course Topics:</b></p> <ul style="list-style-type: none"> <li>▪ Java Environment</li> <li>▪ Java Basics</li> <li>▪ Classes in Java</li> <li>▪ Arrays and Strings</li> <li>▪ Applying Inheritance</li> <li>▪ Writing Java Applets</li> <li>▪ More of the Java API</li> <li>▪ Abstract Windowing Toolkit</li> <li>▪ Exceptions</li> <li>▪ Collection classes</li> <li>▪ Threads</li> <li>▪ I/O Streams</li> <li>▪ Networking</li> <li>▪ Testing with JUnit</li> <li>▪ Persistence</li> <li>▪ Database access with JDBC</li> </ul>	<p><b>Audience:</b></p> <ul style="list-style-type: none"> <li>▪ Programmers new to the Java Language</li> <li>▪ Programmers converting from a non-OO environment</li> <li>▪ Designers needing a review of OO and Java technology</li> </ul> <p><b>Prerequisites:</b></p> <ul style="list-style-type: none"> <li>▪ Programming experience</li> <li>▪ An understanding of object-oriented concepts will be useful but not it is essential.</li> <li>▪ Awareness of software development</li> </ul>

<p>V2.2</p>	<p style="text-align: right;">Oak Lodge Consulting Ltd Cambridge 01223-890390 <a href="http://www.oaklodgeconsulting.co.uk">www.oaklodgeconsulting.co.uk</a></p>
-------------	--

LN 029  
5 days

# Java™ Programming

## Course Outline:

### Java Environment

- 📁 Java run-time Environment
- 📁 Developing Applications
- 📁 Classes, Files, Directories

### Java Basics

- 📁 Variables, Data Types, Statements, Flow Control
- 📁 Simple Output

### Arrays and Strings

- 📁 Defining arrays
- 📁 String Objects and Methods
- 📁 String Conversion

### Object Orientation in Java

- 📁 Instantiating a Class
- 📁 Member Access
- 📁 Method Overloading
- 📁 Constructors
- 📁 Encapsulation
- 📁 Inheritance
- 📁 Polymorphism

### Applying Inheritance

- 📁 Derivation Syntax
- 📁 Effects of Inheritance
- 📁 Overriding Methods
- 📁 Abstract Classes
- 📁 Interfaces
- 📁 Packages

### Exceptions

- 📁 Java Exceptions throw, try catch, finally and throws
- 📁 Exception Types

### Abstract Windowing Toolkit and Events

- 📁 Components
- 📁 Action Events
- 📁 Adapter Classes
- 📁 Inner Classes
- 📁 Text Components
- 📁 Using the AWT and Swing
- 📁 Frames and Menus
- 📁 Standalone Windows Applications

### Writing Java Applets

- 📁 Applet Characteristics and Security
- 📁 Graphics, Fonts, Lines, Shapes, and Colour
- 📁 Events and Java Event Models
- 📁 Using, Locating, Loading, and Displaying Images
- 📁 Animation and Multi-threading

### Java Testing

- 📁 JUnit overview
- 📁 Test Strategy
- 📁 Delivering Testable code

### More of the Java API

- 📁 Object Class
- 📁 Collections
- 📁 Wrapper Classes
- 📁 Extensions to Java API
- 📁 Documentation and Jar files

### I/O Streams

- 📁 Input Stream and Output Stream
- 📁 Memory and Filter Streams
- 📁 Buffered I/O
- 📁 Reader and Writer
- 📁 Serialization

### Threads

- 📁 Class Thread and Thread Synchronization

### Networking

- 📁 Sockets
- 📁 Client / Server Applications

### JDBC

- 📁 Database
- 📁 Persistence
- 📁 Java and databases