Java Programming for Beginners

Duration:	4 days
Type:	beginner

Description

This is an introductory course for novice developers who have little or no previous programming experience. The course introduces the fundamentals of Java with particular emphasis on basic OO concepts and writing efficient and maintainable code.

Prerequisites

Ideally delegates should have some previous programming experience and knowledge of basic IT concepts.

List of Modules

Core Programming Concepts

- Variables, functions and libraries
- Compiled programs verses interpreted scripts
- Typed and untyped programming
- Static verses dynamic typing
- Threads and call stacks
- Errors and exceptions

Core Java Concepts

- Origins and goals of the Java language
- Bytecode and the Java Virtual Machine
- Packages and dynamic class loading
- Garbage collection in Java

Creating Simple Programs

- Writing Hello World! in Java
- Declaring variables to hold numbers
- Declaring variables to hold strings
- Casting and converting between types
- Reading and writing from the console

Making Choices and Looping

- Conditional execution using the *if* statement
- Conditional execution using the *switch* statement
- Bounded iteration using the *for* loop
- Unbounded iteration using the *while* and *do ... while* loops
- Alerting the execution of a loop using *break* and *continue*
- Different ways of writing infinite loops

Java Methods in Depth

- What does the *static* keyword mean?
- Understanding how parameters are passed
- Choosing if a method should be *public* or *private*
- Understanding recursion and writing recursive methods

Introduction to Object Orientation

- What is a data structure?
- Why are data structures dangerous?
- What are classes and objects?
- What makes an object different from a class?
- Key principle one - Abstraction
- Key principle two - Encapsulation
- How do you avoid repetition in classes?
- Key principle three - Inheritance
- How do you change inherited behavior?
- Key principle four - Polymorphism

Support for OO in Java

- Allocating objects from the heap
- Linking to objects via references
- Moving references to other objects
- Pointing a reference at *null*
- Ensuring an object is no longer referenced
- A brief introduction to the theory of garbage collection

Built in Objects in Java

- Strings are objects in Java
- Arrays are objects in Java
- Every class has an instance of *Class*

Object Oriented Development Part One

- Creating basic Java classes
- Choosing accessibility levels
- Overloading and overriding methods
- Overriding the *toString* method
- Comparing references using the *instanceof* operator
- Comparing references using class objects
- Inheriting from a base class
- Creating abstract and final classes

Object Oriented Development Part Two

- Writing appropriate class constructors
- Private constructors and singletons
- Static and instance initialization blocks
- Top down class and object initialization
- Declaring and using interfaces
- Using inner and anonymous classes
- Implementing *equal* and *hashcode*
- Cloning and copy constructors

GUI Development in Java

- How the Swing library works
- Creating a window class via inheritance
- Adding widgets to the window
- Positioning widgets using layout managers
- Adding event handling to the GUI

Exception Handling in Java

- Introducing errors, runtime exceptions and checked exceptions
- Exceptions in constructors and finalizers
- Implementing an effective exception handling strategy
- The correct use of finally blocks
- Using Java 1.4 assertions

Java I/O

- Introducing input and output streams
- Introducing readers and writers
- Using *File* objects to resolve file paths
- Text based file I/O with buffered streams
- Binary based file I/O using data streams
- Serializing Java objects
- Customizing Java Serialization