Java Essentials

An Introduction to Java Programming for Programmers

For the developer, the Java language is the centrepiece of Sun's Java platform. It is built on a platform independent runtime and forms the foundation for common libraries and middleware.

The *Java Essentials* course is an introduction to the language for experienced programmers who wish to acquire Java language skills. It develops the concepts and syntax through lectures, discussion and hands-on lab exercises.

Objectives

- Introduce the Java language from a modern object-oriented perspective
- Understand object-oriented programming and design in Java
- Implement common patterns in Java
- Emphasise good practice and outline idioms for safe and sensible use of language features

Audience

The course is suitable for programmers who have experience of a modern programming language and at least some familiarity with object-orientation.

Content

The Java Platform The JVM and bytecode · Deployment · The JDK and J2SE · J2SE 5.0 versus earlier versions · Applications, applets, servlets and others · Designing with Java in mind

A Tour of Java Statements, blocks and scope · Comments · Classes, methods and fields · The *main* method · Packages · Basic I/O · Writing and running a program

The Type System Built-in types · Type conversions · Declarations and initialisation · Object types, references and instantiation · The *null* reference · *String* · Arrays · Metadata

Control Flow *if else* and *switch* selection · *while* and *do* loops · Traditional and enhanced *for* loops · Exception handling · Exception safety

Classes and Objects Methods and overloading · *public*, *private* and package visibility · The *this* reference

Initialisation and Finalisation Field initialisation · Initialiser blocks · Constructors · *final* and *static final* fields · The garbage collector · *finalize* and its pitfalls

Interfaces Interfaces and polymorphism · Implementing interfaces · Designing with interfaces · Multiple interfaces · Interface extension · Casting, *instanceof* and guidelines on their use

Extending Classes Single inheritance of implementation · Abstract classes · *protected* visibility · Accessing and initialising the superclass · *final* methods and classes

Exception Objects Checked exceptions, unchecked exceptions and errors · *finally* gotchas

Strings Manipulating and comparing strings · String conversions to and from other built-in types · The *toString* method · String performance pitfalls · *StringBuffer* and *StringBuilder*

Collections The collections API · Overview of generics in Java · Iterators · Using built-in wrappers to store primitive values · Autoboxing and unboxing · Collection wrappers and utilities

Class Scope *static* methods and fields · static top-level classes · Named and anonymous inner classes · Member and local inner classes · *static* import · *enum* types

Library Overview File handling · Threading and synchronisation · Reflection · Applets · The AWT and Swing · Events and listeners · JDBC · Network programming · RMI · JNI · J2EE

Additional Details

Duration 4 days

Setup Projection facilities for a laptop · One workstation per delegate with Java compiler installed (configuration to be agreed) · Whiteboards and/or flip charts

Contact Kevlin Henney · kevlin@curbralan.com · Curbralan Limited · +44 117 942 2990