

Fundamentals of Java

Level: beginner

Length: 35 hours

Course Objective: teach the Java programming language, teach the main principles and mechanisms of object oriented programming, how they are supported and used in Java

What you will learn

- How to develop applications using the Java programming language
- How to use an integrated development environment like Eclipse
- You will learn the syntax, how Java supports the object oriented programming paradigm, the Java virtual machine and standard packages.
- The attendees will exercise their communication & presentation skills, will learn to work together in small teams.

Who Can Attend

- Anybody with a basic knowledge of programming and familiar with using an operating system (like Windows or Linux)
- Programmers who come from procedural programming languages world, for example C, and want to learn Java
- Programmers who already know an object oriented programming language like C++ or C# and want to learn Java

Required facilities: VGA projector, white board, workstation, Oracle JDK, Eclipse or other Java development environment.

Related courses: Design Patterns, Fundamentals of UML, Java Enterprise

Description

The course is focused on the Java language and on the main class packages. The major part of the time budget is dedicated to exercise the language, to learn the particularities and stereotypes used in Java, the practical aspects of object oriented programming.

Note: the course is personalized to follow the attendee's profile, the approaching style depends on their expertise and experience.

Contents

1. Introduction in Java programming: classes and objects, instance and static members, inheritance and aggregation
2. Language fundamentals: basic elements, primitive data types, variable declarations, initial values for variables
3. Declarations: class declarations, methods declarations, constructors, enumerated types, arrays, parameter passing, main()
4. Access control: java source file structure, packages, jar utility, system properties, scope rules, accessibility modifiers, modifiers
5. Operators and expressions: conversions, precedence and associativity for operators, operator precedence and associativity, operators
6. Control flow: statements, exceptions, assertions
7. Object oriented programming: inheritance, OOP concepts, polymorphism, upcasting and downcasting, methods overriding, variables hiding, interfaces, constructors linking
8. Nested type declarations: static member types, non-static member classes, local classes, anonymous classes
9. Object life cycle: garbage collection, object finalization, initializes, object initial state
10. Fundamental classes: java.lang package, Object class, wrapper classes, String class, StringBuilder & StringBuffer
11. Files and streams: File class, byte and character streams, filters, objects serialization
12. Localization, pattern matching, formatting: java.util.Locale class, java.util.Date class, java.util.Calendar class, java.text.DateFormat class, java.text.NumberFormat class, regular expressions, formatting
13. Threads: java.lang.Runnable, java.lang.Thread, synchronization, thread states, priorities, starting/stopping, sleep/wake up, wait/notify, join, blocking types, termination, deadlock
14. Generics: generic types and parameterized types, collections & generics, wildcards, references of wildcard parameterized types, bounded type parameters, generic methods and constructors
15. Collections and maps: interfaces, implementations, collections, sets, collection personalization, particular issues