

Course : Developing from C++ to Java

Practical course - 5d - 35h00 - Ref. LJV

Price : 3070 CHF E.T.

Intensive course providing in-depth mastery of the Java language, applying the principles of Object-oriented programming. After having assimilated the "core language" and the contribution of the different versions of the language, you will use the main libraries, zoom in on input/output concepts and access databases, notably with the JDBC API.



Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Implementing the principles of Object-Oriented Programming
- ✓ Master the syntax of the Java language
- ✓ Master the main standard Java libraries
- ✓ Master an integrated development environment for programming in Java

Intended audience

Designer, developer, engineer, operational project manager.

Prerequisites

Good knowledge of C or C++ programming. Experience in software development required.

Practical details

Hands-on work

The practical exercises have been designed to illustrate all the elements of the language and to systematically implement the concepts of Object-Oriented Design: all exercises include an analysis/design phase followed by a programming phase.

Course schedule

PARTICIPANTS

Designer, developer, engineer, operational project manager.

PREREQUISITES

Good knowledge of C or C++ programming. Experience in software development required.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.

Participants also complete a placement test before and after the course to measure the skills they've developed.

1 Object techniques

- General principles of object modeling and programming.
- Abstraction and encapsulation: interfaces. Different forms of inheritance, polymorphism.
- Introduction to UML modeling: static model, dynamic model, cooperation model, scenarios.

Hands-on work

Application of the concepts to a case study, which will be one of the main themes of the following exercises.

2 Object programming with Java

- Syntax basics: variables, types, expressions, instructions, arrays, control structures and autoboxing.
- Class definition and instantiation. Fields, methods, constructors, static fields and methods.
- Methods with a variable number of arguments. Methodological aspects: class design.
- Compilation units and packages: class visibility control, import mechanism.
- The different forms of inheritance: extension and implementation.
- Interfaces and interface implementation.
- Polymorphism and its implementation.
- The construction of class hierarchies.
- Defining derived classes, constructors and references.
- Code factorization: abstract classes.
- Simultaneous use of implementation and extension.
- Abstract classes.
- Generic types.
- Methodological aspects: grouping constants, specifying services.

Hands-on work

Get to grips with the development environment and program a simple program. Programming the case study. Design and build a hierarchy of classes and interfaces. Implementation of polymorphism and genericity. Introduction of exceptions.

3 Graphical user interface design (AWT, Swing)

- Basic concepts: principles of visualization and event management, some generic classes.
- Visualization of graphic components.
- Containers and Layouts: BorderLayout, FlowLayout, GridLayout, GridBagLayout and CardLayout.
- Building hierarchical containers.
- Some graphic components: labels, buttons, menus, text boxes, checkboxes, canvas.
- Event management. Listeners and Adapters.
- Associating managers with graphic components.
- Swing's special features.

Hands-on work

Build small applications or a small GUI for a few objects in the case study. Build a small editor or event handling GUI for the case study.

TEACHING AIDS AND TECHNICAL RESOURCES

- The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
- At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
- A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

4 Introduction to Web programming: applets

- Web principles and components.
- Applets: principles, life cycle, the Applet class.
- Integrating an Applet into an HTML page, passing parameters, security issues.

Hands-on work

Building an Applet.

5 I/O and some utility classes

- I/O. The hierarchy of I/O classes. Some file system manipulation classes.
- Some input/output classes working on byte streams and char streams. Keyboard input/output.
- Serialization.
- Type encapsulation classes.
- System classes.
- Container classes.
- Observer classes.

6 Connecting to databases : JDBC

- JDBC model and general principles.
- SQL reminder.
- DBMS connection.
- Query execution and response processing.
- The use of precompiled queries.
- Transactional concepts (A.C.I.D.): atomicity, consistency, isolation, durability.

Hands-on work

Saving and consulting data in a database.