

# Java Programming

Hands-on course of 5 days - 35h

Ref.: LJO - Price 2024: CHF2 970 (excl. taxes)

This course will allow you to master the principles of the object approach and the features of the Java language. The language constructions will be introduced progressively starting from basic concepts. The course will also cover design problems (via the UML notation) and will present the main standard libraries and APIs: inputs/outputs, utilities, graphics classes (AWT and Swing), applets. This course briefly presents the new Java Tiger features.

## HANDS-ON WORK

The hands-on exercises were designed to illustrate all elements of the language and to implement the concepts of object-oriented design: All of the exercises include an analysis/design phase followed by a programming phase.

## THE PROGRAMME

last updated: 01/2018

### 1) "Object" techniques

- The general principles of "object" modelling and programming. Abstraction and encapsulation: the interfaces. Various typ
- An introduction to models and to UML notation: static models, dynamic models, cooperation models, and scenarios.

*Hands-on work* : The UML specification for a case study that will be the basis for the exercises that follow.

### 2) An initial language approach

- Variables: declaring and classifying.
- Defining fields.
- Methods: defining.
- Expressions.
- Control instructions: conditional, loop, and branch instructions.
- Tables.
- Compilation Units and packages: controlling the visibility of classes, import mechanisms.

*Hands-on work* : A series of simple exercises allowing familiarisation with the development environment and the realisation of a simple programme. Using packages.

### 3) Defining and instancing classes

- Classes and objects.
- Fields, methods.
- Constructors.
- Self-reference.
- Static fields and methods.
- Aspects of methodology: designing classes.

*Hands-on work* : Programming the case study.

### 4) Inheritance

- The different types of inheritance: extension and implementation.
- Interfaces and the implementation of interfaces. Polymorphism and its utilisation.

## PARTICIPANTS

Developers, engineers, project leaders with close ties to development.

## PREREQUISITES

A base Knowledge in programming. Experience desirable in application's development.

## 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.

## 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.

- Extension. Defining derived classes, constructors, and references. Aspects of methodology. Constructing hierarchies of c
- Simultaneous use of implementation and extension. Abstract classes. Aspects of methodology: Regrouping constants, specif

*Hands-on work : Designing and constructing a hierarchy of classes and interfaces. Implementing polymorphism and genericity in the case study.*

### 5) Exceptions

- The try blocks, generating exceptions.
- The catch selection algorithm ().
- Aspects of methodology: constructing an exception hierarchy, using exceptions.

*Hands-on work : Introducing exceptions into the case study.*

### 6) New Java Tiger features

- The new loop instruction.
- Enumerated types, autoboxing.
- Methods with a variable number of arguments.
- Static imports.
- Generic types.

### 7) Conclusion

- The first assessments on the language.
- The latest trends.
- Bibliography items.

## DATES

---

REMOTE CLASS

2024 : 08 Jul, 07 Oct