Course : Design Patterns, implementation

Practical course - 5d - 35h00 - Ref. DES
Price : 2930 CHF E.T.

4,9 / 5

BEST

This course will train you in application design and modern design practices such as test-driven development and refactoring. Numerous case studies will teach you how to create scalable and reusable applications, taking into account the main design patterns.

Teaching objectives

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

- Understand the fundamentals of object-oriented design
- Apply the basic rules for dividing an application into packages
- Apply the principles of application class construction
- Put test-driven development into practice
- Implementing the main Design Patterns

Intended audience

Designers, developers, architects or project managers.

Prerequisites

Knowledge of an object-oriented language.

Practical details

Hands-on work

Participant workshops will be run in the language of their choice (Python, Java, C++, C# or VB, .Net).

Course schedule

1 Design presentation

- A reminder of the fundamentals of OOP and UML.
- The benefits of UML for design.
- The challenges of design.
- Using inheritance. Advantages and disadvantages.

PARTICIPANTS

Designers, developers, architects or project managers.

PREREQUISITES

Knowledge of an object-oriented language.

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.

2 Fundamentals of object-oriented design

- The open-close principle (OCP) and Liskov substitution principle (LSP).
- Concept of polymorphism, weak coupling and strong cohesion.
- The impact of object-oriented design on projects.

Hands-on work
Division of responsibilities between classes.

3 Class construction principles

- Dependency management with dependency inversion (DIP).
- Reducing apparent complexity by separating interfaces (ISP).
- Division of responsibilities with GRASP.

4 Package organization principles

- The package: a delivery/reuse design unit (REP) and common reuse (CRP).
- Package breakdown. CCP.
- Organization between packages.

Hands-on work
Building and prioritizing packages.

5 Test-driven development

- Test Driven Development (TDD) versus Model Driven Engineering (MDE).
- Writing test cases and test suites.

Hands-on work
Assigning responsibilities to software components using the TDD approach.

6 Design Pattern principles

- Design Patterns for reusing experience.
- Scope, advantages and limitations of Design Patterns.
- Responding to recurring problems.
- The founding patterns of Gamma and GoF: creation, behavior and structure patterns.
- Refactoring. Why refactor?
- Modification of code presentation and class algorithms. Redesign.

Hands-on work
Example of design, refactoring and programming with GoF patterns.

7 Software architecture and architectural patterns

- From requirements to architecture.
- Architectural styles.
- Distribution patterns (Client Server Style, Data Bus Pattern, Blackboard, Repository).
- System design patterns (MVC, layered architecture, Plug-in Style, Pipeline).

8 Development process

- Design in an iterative, incremental process.
- The Agile manifesto. XP, Scrum.

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.

Dates and locations

REMOTE CLASS
2026 : 16 Mar., 18 May, 29 June, 7 Sep., 16 Nov.