Introduction to object-oriented development

Duration: 1 day
Audience: Professionnel TI

Who is this training for?

Analysts, designers, architects, programmers and managers involved in development.

Training objectives

- Understand the terminology and concepts of object-oriented technologies
- Effectively integrate object-oriented notions into your projects
- Compare and explain the differences and similarities between traditional and object-oriented development approaches

Summary

Expose all terms, foundations, concepts, and best practices about the object. This course describes the terminology and concepts of object technologies. He clearly explains with concrete examples how these concepts are effectively integrated into projects.

Course outline

- State of the art in software engineering and evolution of software development processes
- Concepts: objects, classes, components, instances, methods, properties, encapsulation, inheritance, overload, polymorphism, override, and composition
- Differences between development processes (traditional, object-oriented, and agile), phases, and iterations,
- Object persistence and object-relational correspondence
- Principles of object-oriented analysis and design
- Object-oriented languages and environments: C++, Java and C# (origin, purpose, main characteristics and use) and other languages
- Reuse: internal, external, costs, orientations and component libraries
- The importance of UML, restructuring, design patterns and dependency management

Copyright © 2026 AFI by Edgenda inc.

- Advantages and benefits of the object-oriented approach
- Application of all concepts with practical and progressive exercises for a project

Approach and methodology

- Lectures
- Exercises

Recommendations

Good computer literacy or equivalent content.

Copyright © 2026 AFI by Edgenda inc.