Nazzaro & Associates
The Future of Technology Today℠

Course: Object-Oriented Analysis & Design

Goals

The demands of today's marketplace have raised the threshold for object-oriented ability. This course emphasizes the conceptual basis of object-oriented thinking through continuous application of key ideas, teaching students to "think like an object". This course uses the IBM Rational Unified Process® (RUP) and the Unified Modeling Language (UML) to teach object-oriented analysis & design in an example iterative, incremental development process. Extensive hands-on exercises using two complete, and parallel, case studies assure that students see how a concept is modeled, and then have the opportunity to immediately apply and test their understanding.

At the end of the course, the student will be able to:

- Understand core UML notation, and produce essential UML diagrams.
- Effectively model software systems using UML in both analysis and design perspectives.
- Understand how object-oriented modeling is done in an iterative development process.
- Define system requirements through use case techniques.
- Describe and apply the major object-oriented concepts.

Duration

Four or five days.

Prerequisites

Experience in software analysis, design or development is desirable, but not mandatory.

Cost

Please call **1-610-831-1151** for public enrollment and private, on-site pricing.

Description

This course provides students an intensive, practical training in the concepts and application of OO analysis and design. Starting with basic concepts for newcomers, this course challenges students with real-world examples and exercises of OO thinking and UML modeling.

Topics

Object-Oriented Software Process
- Object-oriented vs. procedural approaches
- Iterative development: The Unified Process

The Unified Modeling Language

Transitioning to Objects
- Understanding Classes and Objects
- The 8 Main Concepts

Analysis Modeling
- Discovering domain classes
- Responsibility-driven analysis
- UML Static Modeling
  - Class Relationships
- Evolving the Class Diagram
  - Modeling Traps
  - Detecting Model Mistakes
- UML Dynamic Modeling
  - Interaction & Activity Diagrams
  - State Machine Diagrams

Architecture and Design
- Architectural Styles & Patterns
- Designing for the Web
- Service-Oriented Architecture
- Wrapping Legacy Systems

Design Principles
- Meyer's Open-Closed Principle
- Martin's Design Principles
- Separation of Concerns
- Gang of Four Patterns

Design Modeling
- UML Static Modeling
  - Designing Relationships
- UML Dynamic Modeling
  - Interaction Diagrams
  - Defining Design-level Interfaces
  - Design-level State Machine Diagrams

Persistence Design
- Mapping Objects to RDBMS

Audience

Business or system analysts, technical managers, and software developers who need a common, practical technique for describing and constructing object-oriented systems.

For more information about this course or other courses please contact
Nazzaro & Associates at 1-610-831-1151.