

## Object-Oriented Design with UML

### Objectives

- Discuss role of architecture as it affects designer and developer
- Introduce Model-Driven Architecture® (MDA®)
- Learn how to transform analysis model into design model that is specification for implementation
- Apply the universal process pattern to object-oriented design using UML
- Organize design model; identify subsystems and interfaces
- Apply design patterns using UML collaborations
- Apply collaborations to internal subsystem design

### Description

*Object-Oriented Design with UML* teaches the designer/developer the more advanced techniques to apply the Unified Modeling Language (UML 2.0) to building a design model for a software application. The course begins with an overview of the role of architecture and how it affects design/implementation activities. Detailed architecture design activities are reviewed with an emphasis on packaging and partitioning the design model and identifying subsystems and their interfaces. Design patterns such as the Command, Façade, and Observer patterns from the Gang of Four (GoF) are introduced and applied to architecture and detailed design activities. UML collaborations are used to capture behavior and structure using sequence and class diagrams. Collaborations are applied to deriving the detailed design model from the analysis model for each use case and any identified subsystems. Considerations for performing detailed class design are discussed such as attribute and operation signatures, refining relationships, and multiplicity. On-hands classroom experience allows for successful application of intermediate object-oriented modeling techniques.

### Course Outline (Modules and Topics)

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Architecture Overview               <ul style="list-style-type: none"> <li>• Definition of software architecture</li> <li>• Architectural abstraction, components, structure and behavior</li> <li>• Architecture and design process</li> <li>• The 2+9+1 modeling framework</li> <li>• Review UML collaborations</li> </ul> </li> <li>• Architecture Design               <ul style="list-style-type: none"> <li>• Role of architecture in design</li> <li>• Packaging and partitioning strategies</li> </ul> </li> <li>• Identify Subsystems               <ul style="list-style-type: none"> <li>• Identify subsystems and interfaces</li> <li>• Interface inheritance</li> </ul> </li> <li>• Design Patterns               <ul style="list-style-type: none"> <li>• Model patterns with UML collaborations</li> <li>• Gang of Four (GoF) OO design patterns</li> <li>• Apply patterns to architecture and detailed design</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Design Classes               <ul style="list-style-type: none"> <li>• Applying design principles to analysis classes</li> <li>• Considerations for persistence</li> <li>• Class visibility</li> <li>• Operation visibility, scope, signature, parameters</li> <li>• Attribute visibility, scope</li> <li>• State modeling</li> <li>• Associations, dependencies</li> </ul> </li> <li>• Subsystem Design               <ul style="list-style-type: none"> <li>• Applying universal process pattern to interfaces</li> <li>• Modeling subsystem structure and behavior using collaborations</li> <li>• Determining dependencies between subsystems and other software elements</li> </ul> </li> </ul> |
|---|--|

### Duration

2 days

### Course #

01-0102

### Prerequisites

- Knowledge experience with object-oriented programming languages and development environment/platforms
- Object-Oriented Analysis with UML course

### Continuing education

- Modeling Service-Oriented Architecture
- User Interaction Design with UML
- Test Case Design with UML
- IBM Rational Software Modeling Fundamentals
- Sparx Enterprise Architect Modeling Fundamentals

### Classroom requirements

- No computers required

### Audience

- Software architect
- Software designer
- Developer
- Programmer

### In partnership with

Proven ▶▶▶▶▶ Practical ▶▶▶▶▶ Process™