ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Applying OOAD using UML 1.0

Course Summary

Description

The course begins with a thorough introduction to the fundamental concepts of the object-oriented model and object-oriented programming, and moves into in depth coverage of analysis and design techniques, with special emphasis on design patterns. Students will explore the full system lifecycle from initial conception to final delivery.

Objectives
At the end of this course, students will be able to:

- Understand the Object Oriented Paradigm
- Use UML diagrams for modeling systems
- Use the Unified process to guide the analysis and design of a system
- Use Actors and Use-Cases to drive requirements capture
- Build analysis models
- Evolve the analysis model into a complex component-based architectural model
- Use iterative round trip analysis and design techniques
- Verify "goodness" by applying a set of rules and guidelines.

Topics

- Introduction to Modeling, UML and USDP
- Classes and Objects
- Relationships
- States and Activities
- UML Diagrams
- Use Cases
- Use Case Scenarios
- Conceptual Modeling
- Object Oriented Analysis
- Discovering Potential Objects using CRC Cards
- Static Design Concepts
- Dynamic Design Concepts
- Domain Design
- Detailed Design
- Summary & Conclusion

Audience

This is a beginner level programming course, designed for developers who specify, design and develop software and applications using traditional/formal/structured methods and want to learn to use an object-oriented approach.

Prerequisites

Students should have some working knowledge of a procedural programming language and syntax, such as C.

Duration

Five days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Applying OOAD using UML 1.0

Course Outline

I. Introduction to Modeling, UML and USDP
- A. Building Models
- B. Notation
- C. Domains
- D. The Process of OO Analysis and Design
- E. The Unified Software Development Process

II. Classes and Objects
- A. Objects Provide a Service
- B. Abstractions
- C. Responsibilities and Operations
- D. Messages and Public Interfaces
- E. Instances & Classes
- F. Instantiation
- G. UML Class and Instance Icons
- H. Encapsulation

III. Relationships
- A. Static Relationships
- B. Dependencies
- C. Associations
- D. Navigability
- E. Whole/Part Associations
- F. Composition
- G. Generalization/Specialization Relationships
- H. Inheritance of Methods and Method Overriding
- I. Abstract Classes
- J. Dynamic Relationships
- K. Sequence Diagrams
- L. Collaboration Diagrams

IV. States and Activities
- A. State Diagrams: Object Lifecycles
- B. Definitions
- C. States
- D. Entry and Exit Actions
- E. Activity
- F. Statecharts Model a Single Object
- G. Activity Diagrams

V. UML Diagrams
- A. Class Diagram
- B. Use Case Diagrams
- C. Interaction Diagrams
- D. Sequence Diagrams
- E. Collaboration Diagrams
- F. State Charts; Statechart Diagram
- G. Activity Diagram
- H. Implementation Diagrams

VI. Use Cases
- A. Discovering the Use Cases
- B. Actors
- C. Use Case
- D. Caveats!
- E. Extending Use Cases
- F. Generalizations

VII. Use Case Scenarios
- A. Scenarios
- B. Primary and Secondary Scenarios
- C. Essential and Real Scenarios
- D. Documenting Use Cases and Scenarios
- E. Use Case Benefits

VIII. Conceptual Modeling
- A. Conceptual Modeling
- B. Concepts
- C. Identifying Concepts
- D. Mapmaking Principles
- E. Attributes versus Concepts
- F. Specification or Description
- G. Associations
- H. Common Association List

IX. Object Oriented Analysis
- A. Domain Behavior Modeling
- B. System Sequence Diagrams
- C. Analysis State Diagrams
- D. Contracts

X. Discovering Potential Objects using CRC Cards
- A. Discovering Objects
- B. Brainstorming for Classes
- C. CRC cards & CRC Steps

XI. Static Design Concepts
- A. Visibility of Attributes and Operations
- B. Multiplicity of Objects
- C. Interfaces and Components

Applying OOAD using UML 1.0

Course Outline (cont'd)

D. Design Complex Systems from Components
E. Identifying "Good" Classes
F. Multiplicity of Associations
G. Ternary Relationships
H. Role and Role Names
I. Association Qualification
J. Association Class
K. Whole/Part Associations
L. Extensibility Mechanisms:
M. Abstract Classes
N. Types and Substitutability
O. Polymorphism
P. Packages
Q. Using Packages
R. Component Diagrams
S. Deployment Diagrams

XII. Dynamic Design Concepts
A. Interaction Diagrams
B. Sequence Diagrams
C. Collaboration Diagrams
D. State Diagrams and Business Rules
E. Verifying Completeness
F. Advanced States and Transitions
G. Superstates and Substates
H. Concurrent States
I. Activity Diagrams: Swimlanes

XIII. Domain Design
A. Iterative Development
B. Domain Design
C. Detailed Design
D. Forming the Architectural vision
E. Low Coupling Examined

XIV. Detailed Design
A. Detailed Design Steps
B. Detailed Design Activities
C. Ensuring Low Coupling
D. Patterns In Design
E. Mapping to Databases
F. Mapping to User Interfaces
G. About Frameworks
H. Designing Components and Interfaces

XV. Summary & Conclusion
A. Usage of OO Technology
B. Methodologies and Notation
C. Management Issues
D. The Unified Software Development Process
E. Using Risk to Order the Process
F. Implementation Timetable
G. Reuse