Object Oriented Analysis & Design with UML

Course Summary

Summary

This course starts with an overview of object oriented principles and the Rational Unified Process. It then proceeds on with the requirements of analysis using Use Cases. Students will learn how to use these requirements in order to drive the discovery of an object oriented software architecture in a most interactive way. The emphasis is on how to use the Unified Modeling Language (UML) notation, how to analyze and document user requirements and how to design computer systems that best satisfy requirements. In addition, the following UML diagrams are included; activities, sequence, collaboration, deployment and state diagrams.

Objectives

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

- Learn why Object-Oriented Analysis and Design has become one of the hottest new methods for software engineering
- Explore the virtually unlimited bounds of Object-Oriented Analysis and Design for solving not only large but also smaller complex software problems
- Analyze large complex problems
- Design systems to solve large complex problems
- Make the paradigm shift from structured analysis to object-oriented analysis

Topics

- Introduction to Object-Oriented Analysis and Design using UML
- Object-Oriented Principles
- The Rational Unified Process
- Use Case Driven Analysis
- Modeling Dynamic Aspects of the System
- Identifying System Domain
- Identifying System Behavior
- Refining System Domain through Class
- Understanding Object Lifecycles
- Refining States and Statecharts for Best Case Scenarios
- Completing Analysis and Beginning Design
- Designing Well-Formed Classes
- Designing System Behavior
- Detailing Software Specification
- Separating Interface from Implementation for Improved Design
- Defining System Architecture

Audience

This course is designed for Programmers.

Prerequisites

To gain the most from this course, you need a background in solving problems with computers.

Duration

Four or five days

Object Oriented Analysis & Design with UML

Course Outline

I. Introduction to Object-Oriented Analysis and Design using UML
- A. The Unified Modeling Language (UML)
- B. Course Direction
- C. Analysis and Design
- D. Analysis
- E. Design Map

II. Object-Oriented Principles
- A. Object-Oriented Programming Fundamentals
  What is an Object?
- B. Abstraction
- C. Encapsulation
- D. Inheritance
- E. Aggregation
- F. Objects Association
- G. Classes vs. Objects
- H. What is a Class?
- I. Polymorphism

III. The Rational Unified Process
- A. Unified Process Lifecycle
- B. Inception
- C. Elaboration
- D. Construction
- E. Transition
- F. Incremental Iterative Approach

IV. Use Case Driven Analysis
- A. System Scope
- B. Requirements Specification
- C. Origin of Use Case
- D. Use Case's Driven Analysis and Design
- E. Actors
- F. Use Case
- G. Managing Complexity
- H. Packaging Use Cases
- I. Classifying Actors
- J. Writing a Use Case
- K. Developing Use Cases

V. Modeling Dynamic Aspects of the System
- A. Activity Diagrams
- B. Activities
- C. Transitions
- D. Decision Activity
- E. Parallelism
- F. Swimlanes

VI. Identifying System Domain
- A. The Class Diagram
- B. Class
- C. Attributes
- D. Operations
- E. Object Rendering
- F. Identifying Classes
- G. Data Dictionary
- H. Defining Classes
- I. Detail

VII. Identifying System Behavior
- A. Scenarios and Sequence Diagrams
- B. Scenarios
- C. Keep Analysis Logical
- D. Developing Scenarios
- E. Sequence Diagram
- F. Timing
- G. Collating
- H. Controllers
- I. When to Stop
- J. Generating Associations
- K. Generating Operations

VIII. Refining System Domain through Class
- A. Diagram Relationships
- B. Multiplicity
- C. Association and Object Semantics
- D. Constraints and Notes
- E. Association Classes
- F. Aggregation
- G. Propagation Notation
- H. Inheritance
- I. Abstract Classes
- J. IS-A and Substitutability
- K. AND Generalization

IX. Understanding Object Lifceycles
- A. State Definition
- B. States and Events UML Statechart Diagrams
- C. Identifying States and Events
- D. Statechart Diagram
- E. Alternative Courses

Object Oriented Analysis & Design with UML

Course Outline (cont'd)

F. Events and Behavior
G. Action-Expressions
H. Sending Events to Other Objects
I. Entry/Exit Actions
J. Guard Conditions
K. Automatic Transitions
L. Internal Actions
M. Self Transitions

X. Refining States and Statecharts for Best Case Scenarios
A. Composite States
B. Concurrency within a State
C. Concurrency Synchronization
D. History State

XI. Completing Analysis and Beginning Design
A. Analysis is...
B. Test Analysis Model
C. What's Missing?
D. Design
E. Detailed Behavior
F. Classes
G. Attributes and Operations
H. Associations
I. Interfaces
J. Architecture

XII. Designing Well-Formed Classes
A. Well-Formed Classes
B. Attributes and Operations

XIII. Designing System Behavior
A. Collaboration Diagram
B. Collaboration Process
C. Notation
D. Objects and Object Collections
E. Message Sequencing
F. Messages
G. Object Creation and Deletion
H. Conditional Branching
I. Message Repetition
J. Method Signatures
K. Qualifiers
L. Links
M. Updating the Class Diagram

XIV. Detailing Software Specification - Part I
A. Specification
B. Attribute or Association?
C. Design Process
D. Derived Attributes
E. Visibility
F. Attribute Type
G. Operations
H. Class Attributes and Operations
I. Detailing Software Specification - Part II
J. The Need to Specify
K. Association Navigability
L. Navigation and Responsibility
M. Design Process
N. Multiplicity Constraints
O. Implementation
P. Reference Mechanisms
Q. Aggregation

XV. Separating Interface from Implementation for Improved Design
A. What are the Parts of a Class?
B. Well-Formed Classes
C. What is an Interface?
D. Why Interfaces?
E. Alternative Notation
F. Interface Inheritance
G. Interfaces
H. Parameterized Classes

XVI. Defining System Architecture
A. Logical Software Architecture
B. Packages
C. Visibility
D. Package Design
E. Physical Software Architecture
F. Components
G. Hardware Architecture
H. Deployment Diagram
I. Software Deployment

XVII. Completing Design
A. Analysis Deliverables
B. Concluding the Design Phase
C. Design Deliverables
D. The Data Dictionary
E. Moving into Implementation