Object-Oriented Analysis and Design Series Course:
Object-Oriented Analysis and Design (March 2002)

Object-Oriented Analysis and Design is a 30-hour course that teaches object-oriented analysis and design techniques using UML in the context of the Unified Software Development Process. The course provides an introduction to object-oriented theory and the software development life cycle. Students learn proper analysis and design procedures, and their roles in the development process. Students gain hands-on experience with all phases of the development process: requirements, analysis, design, construction, and testing.

Topics

Section I — The Object Paradigm

Introduction to Software Engineering
Evolution of Software Engineering
Object Technology
Advantages and Disadvantages of an Object-Oriented Approach
Analysis and Design

Understanding the Object Paradigm
Classes
Encapsulation
Abstraction
Object Relationships
Application Objects

Inheritance and Polymorphism
Inheriting Attributes and Methods
Extending Components with Inheritance
Polymorphism
Abstract Classes
Multiple Inheritance
Subtyping vs. Subclassing

Software Development Life Cycle
Software Development Life Cycle Model
Waterfall Life Cycle Model
V-Shaped Life Cycle Model
Incremental Life Cycle Model
Spiral Life Cycle Model

Section II — Tools of Analysis and Design

The Unified Software Development Process
Software Development Process
The Unified Process
Life Cycle of the Unified Process
Cycles, Phases, Iterations, Core Workflows, and Workflows and Iterations

Unified Modeling Language (UML)
Models
Views

Computer-Aided Software Engineering (CASE)
Introduction to CASE
Selecting a CASE Tool

Section III — The Requirements Workflow

The Requirements Workflow
Introduction to Requirements Capture
Activities of the Requirements Workflow
Requirements and the Unified Process

Use-Case Modeling
Introduction to Use-Case Diagrams
Developing a Use-Case Model

Activity Diagrams
Branches and Merges
Forks and Joins
Library System Activity Diagrams

Interface Design and Prototyping
User Interface Design
User Interface Ergonomics
User Interface Prototyping
Specifying System Interfaces

Section IV — The Analysis Workflow

The Analysis Workflow
Introduction to Analysis
Analysis Classes
Use-Case Realizations – Analysis
Analysis and the Unified Process

Analysis Modeling
Collaboration Diagrams
Flow of Events and Special Requirements
Class-Responsibility-Collaboration Cards
Class Analysis

Section V — The Design Workflow

The Design Workflow
Design Model
Use-Case Realizations – Design
Design and the Unified Process

Architecture Modeling
Package Diagrams
Deployment Diagrams

Class Diagrams
Association, Aggregation, Composition, and Generalization
Check Out Asset Class Diagram

Sequence Diagrams
Return Values, Message Conditions, Deletion, Multiplicity, and Return Stack

Statechart Diagrams
States, Transitions, and Superstates

Design Quality Issues
Elements of Good Design
Object-Oriented Design Metrics
Chidamber and Kemerer
Metrics for Object-Oriented Design
Designing for Reuse

The Model-View-Controller Paradigm (MVC)
Model-View-Controller
Building MVC Classes
Hybrid MVC

Refactoring
Identifying New Methods
Identifying Methods That Can Be Moved, Inheritance Opportunities
Clarifying Variable Names

Section VI — The Implementation Workflow

The Implementation Workflow
Identifying Components
Integration Build Planning
Generating Code from Design Classes
Unit Testing
Implementation and the Unified Process

Section VII — The Test Workflow

The Test Workflow
Test Cases, Procedures and Components
Test and the Unified Process

Object-Oriented Analysis and Design Series Course:
Object-Oriented Analysis and Design (March 2002) Continued

Target Audience

Database developers and administrators, Internet application developers, middleware programmers, Java developers, and client/server developers.

Job Responsibilities

Develop n-tier database and legacy connectivity solutions for Web applications using Java, Java APIs, Java Database Connectivity solutions, middleware tools, and distributed object models.

Prerequisites

Students must have CIW Foundations certification or equivalent experience, and have completed the Java Programming series (*Java Programming Fundamentals*) or have equivalent experience with an object-oriented programming language.

Duration

30 hours