

**Course Title: Object-Oriented Analysis and Design Using Rational products**  
**Duration : 4 days**

This four-day program covers the concepts and best practices of software development using object-oriented analysis and design. It includes an overview of the software development life cycle, a detailed coverage of the Unified Modeling Language (UML) version 2.0, and case studies to understand and apply the practices of analysis and design with the object technology.

**Program Objectives**

Some of the key topics covered in the program include:

- An overview of the OO concepts
- Understanding the development life cycle
- Understanding the role of UML in the development life cycle
- Principles and practices of object-oriented analysis
- Principles and practices of object-oriented design
- Introduction of design patterns

**Audience**

This program is intended for experienced software professionals who are involved in systems design, or are currently working as developers but are preparing themselves for / being groomed for playing the role of designers.

The participants are expected to fulfill the following prerequisites:

- At least one year of programming experience
- Programming experience in any object-oriented programming language (e.g., Java, C++, VB.NET, C#, Delphi) for at least six months
- Basic understanding of the OO concepts, such as classes, objects, inheritance, polymorphism, etc

**Day-wise Break-up**

Day	Module	Topic
	Module 1	Overview of Key Concepts
	Module 2	Concepts of Object Orientation
	Module 3	Designing Classes for Relationships
	Module 4	Requirements Determination
	Module 5	Analysis & Design
	Module 6	Architectural Analysis
	Module 7	Use Case Realization
	Module 8	Identify Design Elements
	Module 9	Identify Design Mechanisms

	Module 10	Use Case Design
	Module 11	Class Design
	Module 12	Data Store Classes
	Module 13	Introduction to Design Principles
	Module 14	Other Diagrams in UML

**Module No 13** Participants will be given orientation to various design principles while they are working on class diagram in a given case study

## Course Outline

### Module 1: Overview of Key Concepts

Iterative software development  
Introduction to UML  
Best Practices of software development

### Module 2: Concepts of Object Orientation

Object Model  
Objects and classes  
General UML Modeling Mechanisms,  
Abstraction, encapsulation and interface

### Module 3: Designing Classes for Relationships

Representing associations relationships in class structures  
Navigability requirements  
Representation of association classes  
Composition relationships  
Dependency relationships  
Realization

### Module 4: Requirements Determination

Use case Model  
Actors and use cases  
Use case specifications  
Supplementary Specifications

Activity diagrams  
Glossary  
Case Study Initiation

## **Module 5: Analysis and design**

Key Concepts  
Difference between analysis and Design  
Workflow details

## **Module 6: Architecture Analyses**

Architectural analysis overview  
Define high-level organization of Subsystems  
Identify Analysis Mechanism

## **Module 7: Use Case Analysis Realization**

Find Classes from Use Case behavior  
Types of classes: Entity, Boundary, Data Store, Controller classes  
Distributing use case behavior to objects  
Responsibilities of classes  
Describe attributes and associations  
Analysis Class Diagram  
Sequence diagrams  
Collaboration diagrams  
Map Classes to Mechanism

## **Module 8: Identify Design Elements**

Map Classes to Mechanism  
Identify Design Elements Overview  
Identify Classes and Subsystems  
Identify Subsystems Interfaces

## **Module 9: Identify Design Mechanism**

Identify Design Mechanisms Overview  
Document Architectural Mechanisms  
Documenting Patterns

## **Module 10: Usecase Design with design Mechanism**

Use Case realization with Design mechanisms  
Use Case realization with interface  
Use Case realization with persistence  
Use Case realization with other mechanism

## **Module 11: Class Design Usecase Design with design Mechanism**

Use Case realization with Design mechanisms  
Design Class diagram  
Define Operations  
Define Methods  
Define Associations

## **Module 12: Introduction to Design Principles**

Introduction to design principles  
    Open Close principle  
    The Liskov Substitution Principle  
    Dependency Inversion Principle (DIP)

## **Module 13: Data Store Classes**

Mapping class structures to table structures  
Designing data store classes

## **Module 14: Other Diagrams in UML**

Component diagrams  
Deployment diagrams  
Package diagrams