

**Course Title : Object-Oriented Analysis and Design** **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

**Set-up Requirements**

Participants' machines should have the following installed:

- Any one of the following OO case tools: Enterprise Architect, Borland Together Architect, Rational Rose (if you are using any other OO case tool, please check with us)
- Microsoft Word or any other word-processing software

**Day-wise Break-up**

Day	Module	Topic
Day 1	Module 1	Overview of Key Concepts
	Module 2	Requirements Management
	Module 3	Domain Modeling
	Module 4	State Modeling
Day 2	Module 5	Identifying Classes and Responsibilities

	Module 6	Use Case Realization
Day 3	Module 7	Class Design
	Module 8	Designing Classes for Object Relationships
	Module 9	Designing Classes for Class Relationships
Day 4	Module 10	Inheritance and Delegation
	Module 11	Overview of Design Patterns
	Module 12	Data Store Classes
	Module 13	Other Diagrams in UML

## Course Outline

### Module 1: Overview of Key Concepts

- Objects and classes
- Abstraction, encapsulation and interface
- Iterative software development
- Introduction to UML

### Module 2: Requirements Management

- Actors and use cases
- Use case diagrams
- Use case specifications

### Module 3: Domain Modeling

- Activity diagrams
- Identifying conceptual classes
- Class diagrams
- Association and aggregation relationships
- Association classes
- Generalization relationships

### Module 4: State Modeling

- Understanding the states of an object
- State machine diagrams
- Nested states

### Module 5: Identifying Classes and Responsibilities

- Types of classes: Entity, Boundary, Data Store, Controller classes
- Segregation of responsibilities

### Module 6: Use Case Realization

- Distributing use case behavior to objects
- Sequence diagrams
- Communication diagrams
- Interaction frames

### Module 7: Class Design

- Basics of class design
- Designing attributes
- Designing operations

### Module 8: Designing Classes for Object Relationships

- Representing associations relationships in class structures
- Navigability requirements
- Qualified associations
- Representation of association classes
- Law of Demeter
- Composition relationships
- Dependency relationships
- Object diagrams

### Module 9: Designing Classes for Class Relationships

- Generalization relationships
- Polymorphism
- Abstract classes and interfaces

### Module 10: Inheritance and Delegation

- Delegation for multiple behavior reuse
- The Liskov Substitution Principle
- Programming to an interface
- The fragile derived class
- When not to use inheritance

### Module 11: Overview of Design Patterns

- Introduction to design patterns
- The Adapter design pattern

### Module 12: Data Store Classes

- Mapping class structures to table structures
- Designing data store classes
- Implementing data store classes
- Trade-offs in O-R mappings
- The Proxy design pattern

### Module 13: Other Diagrams in UML

- Component diagrams
- Deployment diagrams
- Package diagrams
- Composite Structure diagrams
- Timing diagrams
- Interaction Overview diagrams