Pragati Software Pvt. Ltd.

QUALITY, AGILITY, ADAPTABILITY

Corporate IT Training House

Course Title: Object-Oriented Analysis and Design — Duration : 4 days

This four-day program covers the principles and best practices of software development using object-oriented analysis and design. It includes coverage of the Unified Modeling Language (UML) version 2, and case studies to understand and apply the concepts to real-world scenarios.

Program Objectives

Some of the key topics covered in the program include:

- Principles and practices of object-oriented analysis
- Introduction to architecture
- Principles and practices of object-oriented design
- A few selected 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:

- Programming experience in any object-oriented programming language for at least two years
- Basic understanding of the OO concepts, such as classes, objects, inheritance, polymorphism, etc

Set-up Requirements

Participants' machines should have any one of the following OO case tools installed: Enterprise Architect, Rational Rose, Microsoft Visio (if you are using any other OO case tool, please check with us).

Day-wise Break-up

Day	Module	Topic
Day 1	Module 1	Domain Modeling
	Module 2	State Modeling
	Module 3	Designing Classes for Relationships
Day 2	Module 4	Introduction to architecture
	Module 5	Use Case Realization
	Module 6	Class Design
Day 3	Module 7	Inheritance and Delegation
	Module 8	Introduction to Design Patterns
Day 4	Module 8	Introduction to Design Patterns (contd.)
	Module 9	Data Store Classes

Course Outline

Module 1: Domain Modeling

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

Module 2: State Modeling

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

Module 3: Designing Classes for Relationships

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

Module 4: Introduction to Architecture

- Overview of architecture
- Model-view separation
- Types of classes: Entity, Boundary, Data Store, Controller classes
- Component diagrams
- Deployment diagrams
- Package diagrams

Module 5: Use Case Realization

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

Module 6: Class Design

- Basics of class design
- Designing attributes
- Designing operations

Module 7: Inheritance and Delegation

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

Module 8: Introduction to Design Patterns

- Introduction to design patterns
- Factory Method pattern
- Abstract Factory pattern
- Singleton pattern
- Façade pattern
- Adapter pattern
- Observer pattern

Module 9: 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