OBJECT-ORIENTED ANALYSIS & DESIGN USING THE UNIFIED MODELING LANGUAGE (5 Day)

Course Description: Learn how to use Object-Oriented techniques to analyze real-world requirements and to design solutions that are ready to code. Students learn how to identify and design objects, classes, and their relationships to each other, which includes links, associations, and inheritance. A strong emphasis is placed on diagram notation for use cases, class and object representation, links and associations, and object messages. This course utilizes UML 2.0 notation.

Audience: Analysts, designers, and programmers responsible for applying OO techniques in their software engineering projects.

Prerequisites: Familiarity with structured techniques such as functional decomposition is helpful.

Course Contents

Introduction to Analysis and Design
- Why is Programming Hard?
- The Tasks of Software Development
- Modules
- Models
- Modeling
- Perspective
- Objects
- Change
- New Paradigms

Objects
- Encapsulation
- Abstraction
- Objects
- Classes
- Responsibilities
- Attributes
- Composite Classes
- Operations and Methods
- Visibility
- Inheritance
- Inheritance Example
- Protected and Package Visibility
- Scope
- Class Scope

Advanced Objects
- Constructors & Destructors
- Instance Creation
- Abstract Classes
- Polymorphism
- Polymorphism Example
- Multiple Inheritance
- Solving Multiple Inheritance Problems
- Interfaces
- Interfaces with Ball and Socket Notation
- Templates

Classes and Their Relationships
- Class Models
- Associations
- Multiplicity
- Qualified Associations
- Roles
- Association Classes
- Composition and Aggregation
- Dependencies
- Using Class Models

Sequence Diagrams
- Sequence Diagrams
- Interaction Frames
- Decisions
- Loops
- Creating and Destroying Objects
- Activation
- Synchronous & Asynchronous
- The Objects Drive the Interactions
- Evaluating Sequence Diagrams
- Using Sequence Diagrams

New Models in UML 2.0
- New to UML 2.0
- Composite Structure Diagrams
- Timing Diagrams
- Interaction Overview Diagrams

Use Cases
- Use Cases
- Use Case Diagram Components
- Use Case Diagram
- Actor Generalization
- Include and Extend
- Other Systems
- Narrative
- Template for Use Case Narrative
- Using Use Cases

Process
- Process
- Risk Management
- Test
- Reviews
- Refactoring
- History
- The Unified Process
- Agile Processes

The Project
- Inception
- Elaboration
- Elaboration II
- Construction Iterations
- Construction Iterations - The Other Stuff

Domain Analysis
- Top View – The Domain Perspective
- Data Dictionary
- Finding the Objects
- Responsibilities, Collaborators, and Attributes
- CRC Cards
- Class Models
- Use Case Models
- Other Models
- Judging the Domain Model

Requirements and Specification
- The Goals
- Understand the Problem
- Specify a Solution
- Prototyping
- The Complex User
- Other Models
- Judging the Requirements Model

Design of Objects
- Design
- Factoring
- Design of Software Objects
- Features
- Methods
- Cohesion of Objects