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:** Strong understanding of Object-Oriented concepts is required. Experience designing or programming in an Object-Oriented language is also required.

**Number of Days:** 2 days

---

1. **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

2. **Class Diagrams**
   - Class Diagrams
   - Attributes
   - Attribute Properties
   - Composite Classes
   - Operations and Methods
   - Inheritance
   - Abstract Classes
   - Interfaces with Ball and Socket Notation
   - Visibility
   - Class Scope

3. **Class Diagrams and Their Relationships**
   - Dependencies
   - Associations
   - Instance Creation
   - Multiplicity
   - Qualified Associations
   - Association Classes
   - Composition and Aggregation

4. **Sequence Diagrams**
   - Sequence Diagrams
   - Interaction Frames
   - Decisions
   - Loops
   - Creating and Destroying Objects
   - Activation - 2.0
   - Synchronous & Asynchronous
   - The Objects Drive the Interactions
   - Evaluating Sequence Diagrams
   - Using Sequence Diagrams

5. **Communication Diagrams**
   - Communication Diagrams
   - Communication and Class Diagrams
   - Evaluating Communication Diagrams
   - Using Communication Diagrams

6. **State Machine Diagrams**
   - What is State?
   - State Notation
   - Transitions and Guards
   - Registers and Actions
   - More Actions
   - Internal Transitions
   - Superstates and Substates
   - Concurrent States
   - Using State Machines
   - Implementation

7. **Activity Diagrams**
   - Activity Notation
   - Decisions and Merges
   - Synchronization
Drilling Down
Iteration
Partitions
Parameters and Pins
Expansion Regions
Using Activity Diagrams

8. **New Models in UML 2.0**
New to UML 2.0
Composite Structure Diagrams
Timing Diagrams
Interaction Overview Diagrams

9. **Package, Component, and Deployment Diagrams**
Modeling Groups of Elements – Package Diagrams
Visibility and Importing
Structural Diagrams
Components and Interfaces
Deployment Diagram

10. **Appendix A – UML Syntax**