The Unified Modeling Language

The Unified Modeling Language is an industry-standard method for constructing a model of a software system by visualizing, documenting, and specifying the architecture of the system.

In this course, 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.

Course Objectives:

- Use modeling in analysis and design, particularly in visual modeling.
- Use the Unified Modeling Language to create visual models of business problems and software solutions.
- Create models to show relationships between classes.
- Create models to portray activities performed by objects.
- Create models to portray complex algorithms.
- Create models to show object state.
- Create models to portray object creation.

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 Course Introduction
Course Objectives
The Unified Process
Course Overview
Using the Workbook
Suggested References

2 Use Cases
Use Cases
Use Case Diagram Components
Use Case Diagram
Actor Generalizations
Include and Extend
Specialize
Other Systems
Narrative
Template for Use Case Narrative
Using Use Cases

3 Class Diagrams
Class Diagrams
Attributes
Attribute Properties
Operations and Methods
Inheritance
Abstract Classes
Interfaces
Interfaces with Ball and Socket Notation
Visibility
Class Scope

4 Class Diagrams and Their Relationships
Dependencies
Associations
Instance Creation
Multiplicity

Qualified Associations
Association Classes
Composition and Aggregation

5 Sequence Diagrams
Sequence Diagrams
Interaction Frames
Decisions
Loops
Creating and Destroying Objects
Activation
Synchronous & Asynchronous
Evaluating Sequence Diagrams
Using Sequence Diagrams

6 Communication Diagrams
Communication Diagrams
Communication and Class Diagrams
Evaluating Communication Diagrams
Using Communication Diagrams

7 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

8 Activity Diagrams
Activity Notation
Decisions and Merges
Forks and Joins
Drilling Down
Iteration
Partitions
Signals
Parameters and Pins
Expansion Regions
Using Activity Diagrams

9 Supplemental UML Diagrams
Modeling Groups of Elements – Package Diagrams
Visibility and Importing
Structural Diagrams
Components and Interfaces
Deployment Diagram
Composite Structure Diagrams
Timing Diagrams
Interaction Overview Diagrams

10 Appendix A – UML Syntax