

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

<p>1 Course Introduction Course Objectives Overview Suggested References</p>	<p>2 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</p>	<p>4 Class Diagrams and Their Relationships Dependencies Associations Instance Creation Multiplicity Qualified Associations Association Classes Composition and Aggregation</p>	<p>Attribute Properties Composite Classes Operations and Methods Inheritance Abstract Classes Interfaces with Ball and Socket Notation Visibility Class Scope</p>
<p>3 Class Diagrams Class Diagrams Attributes</p>			

5	<p>Sequence Diagrams</p> <ul style="list-style-type: none"> 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 	<ul style="list-style-type: none"> Visibility and Importing Structural Diagrams Components and Interfaces Deployment Diagram
6	<p>Communication Diagrams</p> <ul style="list-style-type: none"> Communication Diagrams Communication and Class Diagrams Evaluating Communication Diagrams Using Communication Diagrams 	11
7	<p>State Machine Diagrams</p> <ul style="list-style-type: none"> 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	<p>Activity Diagrams</p> <ul style="list-style-type: none"> Activity Notation Decisions and Merges Synchronization Drilling Down Iteration Partitions Parameters and Pins Expansion Regions Using Activity Diagrams 	
9	<p>New Models in UML 2.0</p> <ul style="list-style-type: none"> New to UML 2.0 Composite Structure Diagrams Timing Diagrams Interaction Overview Diagrams 	
10	<p>Package, Component, and Deployment Diagrams</p> <ul style="list-style-type: none"> Modeling Groups of Elements – Package Diagrams 	