



[www.iconATG.com](http://www.iconATG.com)  
+1 866 ICON US1  
[info@iconATG.com](mailto:info@iconATG.com)

## Analysis Using UML 2.0

The features, principles and techniques of object-oriented technology mitigate the complexities of modern software systems. Successful projects have learned that object-oriented programming is insufficient; that object-oriented analysis, architecture, and design are required for robust, scalable, maintainable web-based and conventional business systems, as well as embedded systems. This course teaches the processes, techniques, and artifacts necessary for modern object-oriented analysis. Students will learn the key analysis diagrams, symbols, and concepts of the Unified Modeling Language (UML) v2.0, the de facto international standard for modeling and specifying software systems. UML 2.0 added new diagrams and significantly enhanced older diagrams including class, activity, and state machine diagrams, the focus of analysis modeling. Students will learn through detailed lecture and hands-on labs the core competencies in object-oriented analysis through the use of UML 2.0 diagrams and semantics.

### Objectives:

- < Understand how to identify and classify the objects in business problems and model their business data, behavior, rules and constraints
- < Master the following UML 2.0 diagrams using proven analysis and design methods:
  - Use Case diagrams
  - Activity diagrams
  - Class diagrams
  - Sequence diagrams
  - State Machine diagrams
- < Produce high quality analysis models using system operation contracts
- < Define and practice the processes used in analysis
- < Acquire hands-on experience in these methods and diagrams through case study exercises

### Audience:

This course is designed for business and systems analysts, architects, designers, developers, and testers who develop object-oriented systems. Technical leads and software quality assurance personnel who oversee development of object-oriented systems will also find this course vital.

### Prerequisites:

Prior development experience



## Duration:

3 days

## Outline:

### 1. What are objects?

- Introduce the concept of objects and classes
- Explore roles, tasks, and concepts in using objects to build systems

### 2. What is object-orientation?

- Explore pivotal concepts to create good object oriented systems: cohesion, coupling, abstraction, encapsulation, information hiding, reuse

### 3. How do we do object-oriented work?

- Examine process and artifacts of modern object-oriented methodology
- Introduce the Unified Modeling Language (UML) and its many diagrams

### 4. Requirements: How are requirements documented?

- Review key requirements documents:
  - Vision document
  - Use Case Model
  - Supplementary Specification document
  - Activity Diagrams: activities, control & data flows, decisions, junctions, forks, joins, input & output pins, send signals, receive events, exceptions, interruptible regions, partitions
- Lab: Analyze use case diagrams, activity diagrams, and use case specifications

### 5. Analysis: How do we do analysis?

- Identify step in the analysis process
- Introduce Domain Class Models, System Sequence Diagrams, System Operation Contracts, and State Models

### 6. Analysis: How do we identify domain classes?

- Discuss what things can be objects
- Learn techniques for finding classes of objects
- Lab: Identify classes in use cases

### 7. Analysis: How do we model domain classes?

- Learn the syntax and semantics of Domain Class Diagrams:
  - Classes
  - Attributes
  - Association relationships
  - Generalization relationships
- Lab: Develop domain class diagrams

### 8. Analysis: How do we identify system operations?

- Learn the syntax and semantics of System Sequence Diagrams:
  - The System object
  - System events
  - Focus of control
- Discuss process of drawing System Sequence Diagrams
- Lab: Develop system sequence diagrams from use cases



[www.iconATG.com](http://www.iconATG.com)

+1 866 ICON US1

[info@iconATG.com](mailto:info@iconATG.com)

**9. Analysis: How do we specify system operations?**

- Learn how to specify the System Operations Contracts
- Learn how to specify system operation contract post conditions in terms of changes in domain model state
- Lab: Develop system operations contracts