www.iconATG.com

+1 866 ICON US1
+1 636 530 7776
info@iconATG.com

# Analysis & Design 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 and design. Students will learn the chief 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 composite structure, timing and interaction overview diagrams, and significantly enhanced class, component, sequence, and state machine diagrams. Students will learn through detailed lecture and hands-on labs the core competencies in object-oriented analysis and design 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
  - Component diagrams
  - Composite Structure diagrams
  - Timing diagrams
- Learn and apply key principles that facilitate repeatable, quality designs such as the Liskov Substitution Principle, the Law of Demeter, the Information Expert Principle, High Cohesion, Loose Coupling, and many others
- Understand the basic concepts of object-oriented software architecture including logical partitioning of systems into layers and subsystems, process and thread architecture, and hardware architecture modeling
- Learn what analysis, architecture, and design patterns are and apply them to improve designs
- Acquire hands-on experience in these methods and diagrams through case study exercises

## Audience:

This course is designed for 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.

www.iconATG.com

+1 866 ICON US1
+1 636 530 7776
info@iconATG.com

## Perquisites:

Prior development experience

## Duration:

5 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 objected-oriented methodology
   - Introduce the Unified Modeling Language (UML) and its many diagrams

4. **Requirements: How are requirements documented?**
   - Review key requirements documents and diagrams:
     - 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, timed 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

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

- Learn how to specify the system operations as 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

**10. Architecture: How do we architect a system?**

- Explore the 4+1 view of architecture
- Layering and subsystems
- Component modeling on Component Diagrams
  - Components
  - Provided interfaces
  - Required interfaces
  - Dependencies

**11. Design: How do we design a solution?**

- Learn the steps in the design process
- Move from domain modeling to software modeling

**12. Design: How do we design classes?**

- Learn how to identify software classes
- Refactor classes to enhance cohesion using
  - Delegation to helper classes
  - Model-View-Controller pattern
  - Parts, ports, and connectors on Composite Structure Diagrams
- Lab: Model entity, control, and boundary classes in a design
- Write class specifications
- Apply architectural decisions such as layering
- Lab: Apply a layered architecture to a design

**13. Design: How do we design associations?**

- Learn the full syntax of associations, aggregations, compositions, generalizations, realizations, and dependencies
- Design role visibility and navigability
- Design inheritance and polymorphism in generalization relationships
- Learn the Liskov Substitution Principle
- Lab: Model relationships in a design class model

**14. Design: How do we design attributes?**

- Learn the full syntax of attributes
- Discuss visibility, data typing, multiplicity, class-scope attributes
- Design derived attributes
- Write attribute specifications

**15. Design: How do we identify operations?**

- Learn the syntax and semantics of detailed Sequence Diagrams:
  - Object, activation, creation, and destruction
  - Messages and operation signatures
  - Loop, alternative (alt), option (opt), break & parallel (par) fragments and continuations
  - Interaction use (ref) fragments, decomposed lifelines & gates
  - Critical region fragments
  - Negative (neg), assertion (assert), strict, ignore, and consider interaction operators
  - Part decomposition
- Discuss process of designing object collaboration through sequence diagrams
- Guidelines to promote loosely coupled, highly cohesive designs
- Lab: Develop sequence diagrams

**16. Design: How do we design operations?**

- Learn the full syntax of operations

- Discuss visibility, parameter and return data typing, class-scope operations
- Design operation types: implementer, manager, accessor, and helper
- Enhance design resiliency by applying the Law of Demeter
- Write operation specifications
- Lab: Refactor sequence diagrams and classes to apply the Law of Demeter

**17. Design: How do we model object state?**

- Learn State Machine Diagram syntax and semantics:
  - States, Transitions, Events, Activities, Actions, Guards, Constraints
  - Submachine & composite state entry & exit points
- Apply Timing Diagram semantics to model timed state transitions

**18. Design: How do we use patterns?**

- Define patterns
- Explore representative patterns: Observer, State, Strategy, Abstract Factory
- Learn patterns for all design activities: analysis, architecture, design, and coding
- Modeling patterns and pattern use with Composite Structure Diagrams
- Lab: Apply patterns to a design

www.iconATG.com

+1 866 ICON US1
+1 636 530 7776
info@iconATG.com

# Why IconATG?

- Consulting, mentoring and developing/providing training programs for large IT organizations since 1992
- Full software lifecycle curriculum with cost-effective, tailored courses with seasoned instructors, qualified through hands-on experience
- Skilled in tools selection and deployment, organizational transformation via process, technology and culture
- Proven experience tailoring and extending iterative SDLC processes (RUP, Agile, Scrum, XP, OpenUP, Essential UP)
- Experienced mentors and consultants with demonstrated project success

Consulting & Mentoring
Workshops
Training
Staff Augmentation

2007 Training Stats:

1,153 students
110 courses delivered
85 companies

IconATG is a thought-leader in IT training, consulting and mentoring. Our training team has successfully developed cost-effective, customized IT training programs and we have taught thousands of students through our formal courseware and hands-on workshops. We offer introductory to advanced courses in focused disciplines of the full software lifecycle including Iterative, Agile, Unified Process (RUP), Scrum, UML, requirements and Use Cases, facilitation, user-centered design, iterative project management, and architecture (SOA/MDA). Our instructors' real-world expertise is incorporated in each of our classes, giving your team practical skills to be highly productive when developing today's most demanding applications.

Mentoring solidifies knowledge gained through training by applying concepts learned in class. Icon's extensive project experience has shown that teams better understand new processes and techniques by applying them with an experienced mentor. IconATG consultants and mentors work with project teams applying new technologies, tools and processes in their organizations to ensure project success. Full lifecycle experience allows Icon consultants to deliver expert knowledge in specific disciplines, while providing an understanding of the workflow throughout the lifecycle. We actively work with your project team helping them develop skills and address problems through facilitation, demonstration, co-development, review, observation and advice. IconATG is that critical consulting resource – we help ensure your success!