Analysis using UML

The features, principles and techniques of object-oriented technology mitigate the complexities of modern software systems. This course teaches the processes, techniques, and artifacts necessary for modern object-oriented analysis in an iterative process. This course uses and explains the essential analysis diagrams, symbols, and concepts of the Unified Modeling Language (UML), the international standard modeling language for software. Students will learn through detailed lecture and hands-on labs the core competencies in object-oriented analysis. Labs are performed with or without CASE tools such as Rose as the teaching environment permits.

Objectives:
- Understand how to identify and classify the objects in business problems
- Learn how to model business data, behavior, rules and constraints
- Master UML semantics of class modeling and system sequence diagramming
- Produce high quality analysis models using system operation contracts
- Define and practice the processes used in analysis
- Acquire hands-on experience in these processes, techniques, and Rose CASE tool use through case study exercises

Topics Covered:
- What are objects, classes, and object-orientation
- Roadmap of the iterative object-oriented analysis process
- Class modeling, sequence diagramming, and state modeling
- System sequence diagrams and system operation contracts
- Essential documentation specifications for elements within each of these models
- Optional: Use of the Rose CASE tool to produce class diagrams and system sequence diagrams

Audience:
This course is designed for the systems analysts, architects, designers, developers, and testers who are directly responsible for developing object-oriented systems. This course is also of benefit to technical leads and software quality assurance personnel who oversee development of object-oriented systems and require an understanding of the process and the artifacts being produced.

Prerequisites:
Prior development experience
Duration:
2 days

Outline:

1. **What are objects?**
   - Introduce the concept of objects and classes
   - Explore roles, tasks, and concepts in using objects to build systems
   - Discuss how analysis is performed in an iterative development process

2. **Requirements: How are requirements documented?**
   - Review key requirements documents:
     - Vision document
     - Use Case Model
     - Supplementary Specification document
   - Lab: Analyze use case diagrams and specifications

3. **How do we do analysis?**
   - Identify step in the analysis process
   - Introduce Domain Class Models, System Sequence Diagrams, System Operation Contracts, and State Models

4. **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

5. **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

6. **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

7. **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
   - Refactor domain class diagrams based on system operation contract findings
   - Lab: Develop system operations contracts and refactor domain class diagrams
Why IconATG?

- Thirteen years of experience mentoring, consulting, and developing training programs for large IT organizations
- Full software lifecycle curriculum of cost-effective, tailored courses
- Seasoned instructors qualified through hands-on experience
- Experienced mentors and consultants with demonstrated project success
- Proven experience tailoring and extending the Unified Process and Agile/Scrum Processes

IconATG is a thought-leader in information technology training, mentoring and consulting. Our training staff 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 the Unified Process (RUP), Agile, UML, requirements and use cases, usability, project management and architecture (SOA/MDA). Our instructors’ real-world expertise is incorporated in each of our classes, giving your team the practical skills to be more productive when developing today’s most demanding applications.

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