



Analysis & Design using UML & Use Cases

Explore OO Modeling, OO Process, Use Cases, UML, Best Practices and More

www.triveratech.com

Course Snapshot

- **Course:** Analysis and Design using UML and Use Cases (TT1310)
- **Duration:** 5 days
- **Targeted Audience:** Developers who specify, design and develop software and applications using traditional/formal/structured methods and want to learn how to apply good practices to design and analysis.
- **Course Structure & Hands-On:** Expert lecture and discussions combined with engaging hands-on (pen and paper) Analysis and Design Labs, group discussions and activities. Exercises are “thinking, discussing and drawing” group activities and labs. Student machines are not required.
- **Language / Tools:** This course is programming language independent. This course is also independent of any software development methodology such as waterfall, rapid application development, continuous integration, agile, etc. UML is the primary tool that is used.
- **Delivery Options:** This course is available for **onsite private classroom presentation, live online virtual presentation**, or can be presented in a **flexible blended learning format** for combined onsite and remote attendees. Please also ask about our **Self-Paced / Video** or **QuickSkills / Short Course** options.
- **Customizable:** This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please inquire for details.

Overview

Geared for software analysts, designers and programmers, this five-day course provides a practical approach to producing high quality object-oriented software designs, using some of the most effective techniques in use today, such as Use Case analysis, static and dynamic system modeling, using UML to document designs, and much more.

This is a **beginner** level programming course, designed for developers who specify, design and develop software and applications using traditional/formal/structured methods and want to learn to use an object-oriented approach. Ideally students should have some experience with a procedural programming language and syntax, such as C.

Attendees include systems and software analysts and designers, programmers who read and implement program designs, personnel involved in inspections and design/code walk-through, software project managers managing large (re-use) projects, and maintenance personnel involved in maintaining and re-engineering software products. This course is also highly beneficial for those who specify requirements and business rules for systems.

Learning Objectives

This course includes coverage of the most effective techniques in use today, such as Use Case analysis, static and dynamic system modeling, using UML to document designs, and much more. The focus of the course is to give a practical approach to producing high quality object-oriented software designs and to provide the knowledge and experience necessary to avoid the most common risks associated with building production systems.

Working in a hands-on drawing environment, developers will:

- Learn the three pillars of building a system: The Model, The Process, The Best Practices
- Understand the object-oriented model, including types, objects, encapsulation, abstraction, messaging, protocols, inheritance, polymorphism, relationships, and coupling, strengths and weaknesses
- Understand the importance of a development process, and the risks of not having one, or having a bad one
- Learn how to read and create the most important UML diagrams
- Recognize the difference between analysis and design
- Be able to produce a requirements analysis
- Know how to create Use Cases
- Learn how to create a static conceptual model of your system

- Learn how to create a dynamic behavioral model of your system
- Understand how to move from analysis to design
- Understand the uses of inheritance, where it is appropriate, and where it is not
- Understand the importance and use of interfaces
- Understand how to move from design to implementation
- Discuss testing, test plans, the testing lifecycle and test methodologies

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be easily adjusted to accommodate. We offer additional software development, architecture, programming and design courses which may be blended with this course for a track that best suits your development objectives. Our team will collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Course Case Study

Throughout this course students will explore a "real world", practical project illustration (case study) of a typical application showing all the steps required for requirements capture, analysis, architectural and detailed design. The course week begins with a thorough introduction to the fundamental concepts of the object-oriented model and object-oriented programming and moves into in depth coverage of analysis and design techniques, with special emphasis on design patterns. Students will explore the full system lifecycle from initial conception to final delivery.

Students are provided with a clear set of guidelines and rules that they apply to the modeling, from start to finish, of a typical application. These exercises emphasize all aspects of the modeling process with special attention being paid to reusability, extensibility and complexity management plus other techniques that will increase the likelihood that their projects will succeed.

By exploring the lab Case Study students will learn to

- Understand the Object-Oriented Paradigm
- Know how use UML diagrams for modeling systems
- Use the Unified process to guide the analysis and design of a system
- Use Actors and Use-Cases to drive requirements capture
- Build analysis models
- Evolve the analysis model into a complex component-based architectural model
- Use iterative round-trip analysis and design techniques
- Know how to verify "goodness" by applying a set of rules and guidelines

Audience & Pre-Requisites

This is a **beginner** level programming course, designed for developers or technical managers who specify, design and develop software and applications using traditional/formal/structured methods and want to learn to use an object-oriented approach. Ideally students should have some working knowledge of a procedural programming language and syntax, such as C.

Attendees can include systems and software analysts and designers, programmers who read and implement program designs, personnel involved in inspections and design/code walk-through, software project managers managing large (re-use) projects, and maintenance personnel involved in maintaining and re-engineering software products. This course is also highly beneficial for those who specify requirements and business rules for systems. Attendees should have a working knowledge of developing software applications. Designing and analysis experience is also extremely beneficial. This is not a coding class.

Take After: Our core OO training courses provide students with a solid foundation for continued learning based on role, goals, or their areas of specialty. Our object oriented developer learning paths offer a wide variety of follow-on courses such as:

- Basic level coding courses in OO technologies such as Java, .Net, C++, C#, Python or JavaScript
- Software development, Architecture or Design Patterns courses
- Please contact us for recommended next steps tailored to your longer-term education, project, role or development objectives.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change,

and may adjust during live delivery based on audience interests, skill-level and participation.

Object Oriented Overview

- Object Orientation (OO) defined
- Lab
- OO concepts
- Lab
- Stating the case for OO

The Unified Modeling Language (UML)

- UML defined
- Static diagrams
- Lab: Class Diagram
- Lab: Package Diagram
- Lab: Component Diagram
- Lab: Deployment Diagram
- Dynamic diagrams
- Lab: Collaboration & Sequence
- Lab: state and Activity Diagram

Object-oriented Process Defined

- Software development process overview
- Iterative Process Framework
- Use Case-driven, architecture-centric, iterative and incremental

Object-oriented Analysis and Design

Object-oriented Persistence

Introduction to Use Cases

Use Cases Modeling Concepts

- Actors
- Use Cases
- Applying the Concept: Hands-on Lab / Case Study

Use Cases Modeling Process Defined

- Use Case Modeling Process
- Use Cases
- Applying the Concept: Hands-on Lab / Case Study

Use Cases Modeling Process in Detail

- Use Case Modeling Overview
- Create Initial Use Case
- Detailing Use Cases
- Elaborating Use Cases
- Model Use Case Relationships
- Lab

Use Cases Modeling

- Organizing the Use Case Model
- Use Cases and Business Modeling
- Use Case Instances
- Use Case Testing
- Use Cases and Interface Design
- Use Cases and Analysis Modeling
- Applying the Concept: Hands-on Lab / Case Study

Introduction to OOAD, Modeling, UML and USDP

- Classes and Objects
- Extended Case Study – Labs
- Relationship Lab
- States and Activities
- Applying the Concept: Hands-on Lab / Case Study

Object-Oriented Design

- Static Design Concepts
- Dynamic Design Concepts
- Domain Design
- Applying the Concept: Hands-on Lab / Case Study

Student Materials: Each student will receive a **Student Guide** with course notes, code samples, software tutorials, diagrams and related reference materials and links (as applicable). Our courses also include step by step hands-on lab instructions and solutions, clearly illustrated for users to complete hands-on work in class, and to revisit to review or refresh skills at any time. Students will also receive the project files (or code, if applicable) and solutions required for the hands-on work.

For More Information

Need dedicated training? All courses can be presented **onsite** or **online**, or in a **combined / flex / blended learning format**, tailored to target your specific audience, needs and learning goals. We also offer focused, flexible **short courses**, **self-paced learning** options, **recorded sessions** and more. We train beginner to advanced skills in all areas we cover, and offer **New Hire / Cohort Training**, **Boot Camps**, **Skills Immersion Programs**, **Reskilling Programs**, **Skills Migration & Transition Programs**, and more. We collaborate with you to ensure all courses are truly targeted to meet your specific needs and learning skills, maximizing your valuable training time, as well as your important budget.

Please also visit our extensive **Public Training Schedule** for training for smaller groups or individuals. Please contact us for course details, **Corporate Rates** and **Special Discount Offers**.

For more information about our dedicated training services, collaborative mentoring services, courseware licensing options, courseware development services, public course schedule, training management services, partner and reseller programs, or to see our complete list of course offerings and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**. Our pricing and services are always satisfaction guaranteed.