



Course Outline

TT1130: Applying Object-Oriented Analysis & Design using UML 2.0 (5 days)

Duration: 5 days

Skill Level: Introductory

Focus: Object Oriented Applications; UML design

Format: Expert lecture combined with open discussions and high-Level demonstrations and dynamic group exercises.

Language / Tools: This edition is Language Neutral, although we can easily present geared for Java, .Net or other audiences.

Delivery Format: Available for onsite private classroom presentation, or live online / virtual presentation

Customizable: Yes

The *Applying Object Oriented Analysis & Design using UML 2.0* course is a five-day, comprehensive hands-on workshop that will lay a solid groundwork for any developer to easily move into a Java programming environment. This course takes advantage of several of the new features in UML 2.0

Enterprise application developers today must be able to build, evolve and maintain very complex software solutions. Component-based technologies, such as the Object Oriented software development paradigm, are ideally suited to the development of this type of software. This course provides the techniques necessary to build high quality object-oriented software systems that can fulfill your requirements, can scale in both complexity and capacity, and be more easily understood, extended, and maintained.

Much of the success of this paradigm is due to work of Booch, Jacobson and Rumbaugh who developed the industry standard UML (Unified Modeling Language) and the Unified (initially named Objectory) Process, both based on "best practices" that have been found to work. These tools provide a framework for the analysis, design, programming and testing of software applications. Using the framework provided by UML and the Unified process, the course teaches the student the Object Oriented concepts plus analysis and design techniques and guidelines for modeling complex problems. Models built using these techniques have a very high success rate when turned into working code.

► Course Objectives: What You'll Learn

The course includes coverage of the most effective techniques in use today, such as Use Case analysis, static and dynamic system modeling, responsibility driven design using CRC, Design Patterns, 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 dynamic, interactive, 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 Design Patterns and their importance
- Learn how to apply Design Patterns to refine your model
- 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

The course provides a solid foundation in basic terminology and concepts, extended and built upon throughout the engagement. Processes and best practices are discussed and illustrated through both discussions and group activities.

Attending students will be led through a series of advanced topics comprised of integrated lectures, group discussions and comprehensive demonstrations.

► Course Overview & Structure

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.

All work can be done the old fashioned way – pads, pens and brains - this course can be strictly right brained! However, portions of the work can also be done using some of the commonly available UML tooling (such as those associated with the Eclipse workbench).

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: Who Should Attend

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.

► Related Courses – Suggested Learning Path

Take Instead: We offer other courses that provide different levels of knowledge or focus:

- If you need to work with UML 1.x, you might consider: TT1120 Applying Object-Oriented Analysis & Design Using UML
- If you wish to combine core Patterns essentials, consider TT1250 Object Oriented Design Patterns in Java (or TT1260 - .Net edition)
- If you wish to emphasize design, consider TT1310 Domain Analysis & Design using UML
- Need a faster intro to OO course? Consider TT1100 Core OOAD with UML (3 days)

Take After: We offer a variety of introductory through advanced security, development, project management, engineering, architecture and design courses. Students may want to consider the following topics as a follow-on to this course.

- OO-based programming essentials courses such as Java or .Net fundamentals
- Service-Oriented Analysis and Design
- Web Services – Intro through Advanced
- Architecture & Analysis courses
- Software Engineering, Design or Project Management tracks

Please note all development courses may also be offered in other programming languages or tailored to suit your unique requirements. Please contact us for recommended next steps tailored to your longer term education, project or development objectives.

► Student Materials: What You'll Receive

Our robust course materials include much more than a simple slideshow presentation handout. Student materials include a comprehensive hard-copy course manual, complete with detailed course notes, code samples, diagrams and current reference materials, all directly related to the course at hand, indexed for ease of use. Step-by-step lab instructions and project descriptions are clearly illustrated and commented for maximum learning and ease of use.

Our course kits are designed to serve as an excellent and useful reference set, long after we leave your classroom.



► Experiential Learning: Hands-On Labs

This class is “technology-centric”, designed to train attendees in essential OOAD and UML development skills, coupling the most current, effective techniques with the soundest industry practices.

This workshop is about **50% dynamic lab exercises** and **50% lecture**. Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review. Multiple detailed lab exercises are laced throughout the course, designed to reinforce fundamental skills and concepts learned in the lessons. At the end of each lesson, developers will be tested with a set of review questions to ensure that he/she has fully understands that topic.

► Delivery Environment & Classroom Set Up

Our OOAD courses can be delivered using simply pencil and paper or can be used with UML tooling provided through a variety of tools and IDEs.

Our lab guides are complete with software-specific instructions, screen shots and detailed tutorials for using the software you select. In most cases we can easily port our classes to run in the environment of your choosing.

For course deliveries or virtual presentation using open-source tools, we'll provide our unique **LoadNGo Instant Classroom Kit**, which enables students to run the entire course off of a DVD that hosts the entire course set up software, labs, and other pertinent useful educational resources, whitepapers and more. You only need to provide the hardware and appropriate O/S, and we'll do the rest. No installation needed. **Great for secure environments.** Minimum set up burden for your team or firm, with maximum results for your students.

No matter which set up option or software your firm requires, we're pleased to provide a detailed set up guide for all private or on-site courses, and as much assistance as you require to prepare your students or classroom for the course. Our support personnel and instructors can be contacted for any advice you may require to prepare your classroom and/or students for attendance.

Workshop Topics Covered

Session: Introduction to Modeling, UML and USDP

- Building Models
- Notation
- Domains
- The Process of OO Analysis and Design
- The Unified Software Development Process

Session: Classes and Objects

- Objects Provide a Service
- Abstractions
- Responsibilities and Operations
- Messages and Public Interfaces
- Instances
- Classes
- Instantiation
- UML Class and Instance Icons
- Encapsulation

Session: Relationships

- Static Relationships
- Dependencies

- Associations
- Navigability
- Whole/Part Associations
- Composition
- Generalization/Specialization Relationships
- Inheritance of Methods and Method Overriding
- Abstract Classes
- Dynamic Relationships
- Sequence Diagrams
- Communication Diagrams

Session: States and Activities

- State Diagrams: Object Lifecycles
- Definitions
- States
- Entry and Exit Actions
- Activity
- Statecharts Model a Single Object
- UML 2.0 Activity Diagrams

Session: UML 2.0 Diagrams

- Class Diagram

- Use Case Diagrams
- Interaction Diagrams
- Sequence Diagrams
- Communication Diagrams
- State Machine Diagrams
- Statechart Diagram
- Activity Diagram
- Implementation Diagrams

Session: Use Cases

- Discovering the Use Cases
- Actors
- Use Case
- Caveats!
- Extending Use Cases
- Generalizations

Session: Use Case Scenarios

- Scenarios
- Primary and Secondary Scenarios
- Essential and Real Scenarios
- Documenting Use Cases and Scenarios
- Use Case Benefits



Session: Conceptual Modeling

- Conceptual Modeling
- Concepts
- Identifying Concepts
- Mapmaking Principles
- Attributes versus Concepts
- Specification or Description
- Associations
- Common Association List

Session: Domain Behavior Modeling

- Domain Behavior Modeling
- System Sequence Diagrams
- Analysis State Diagrams
- Contracts

Session: Discovering Potential Objects using CRC Cards

- Discovering Objects
- Brainstorming for Classes
- CRC cards & CRC Steps

Session: Static Design Concepts

- Visibility of Attributes and Operations
- Multiplicity of Objects
- Interfaces and Components
- Design Complex Systems from Components

- Identifying "Good" Classes
- Multiplicity of Associations
- Ternary Relationships
- Role and Role Names
- Association Qualification
- Association Class
- Whole/Part Associations
- Extensibility Mechanisms:
- Abstract Classes
- Types and Substitutability
- Polymorphism
- Packages
- Using Packages
- Component Diagrams
- Deployment Diagrams

Session: Dynamic Design Concepts

- Interaction Diagrams
- Sequence Diagrams
- Interaction Frames
- Communication Diagrams
- Timing Diagrams
- State Diagrams and Business Rules
- Verifying Completeness
- Advanced States and Transitions
- Superstates and Substates
- Concurrent States
- Activity Diagrams: Swimlanes

Session: Domain Design

- Iterative Development
- Domain Design
- Detailed Design
- Forming the Architectural vision
- Low Coupling Examined

Session: Detailed Design

- Detailed Design Steps
- Detailed Design Activities
- Ensuring Low Coupling
- Patterns In Design
- Mapping to Databases
- Mapping to User Interfaces
- About Frameworks
- Designing Components and Interfaces

Session: Summary & Conclusion

- Usage of OO Technology
- Methodologies and Notation
- Management Issues
- The Unified Software Development Process
- Using Risk to Order the Process
- Implementation Timetable
- Reuse

► Why Work With Us?

- **We provide a solid object oriented foundation.** Students will learn how to develop (and reuse!) essential OOAD and UML design skills and concepts properly, using best design practices, grounding them for advanced curriculum. Students will be prepared for designing and implementing real solutions, right after the class ends. Students will learn the importance of developing well-designed OO applications.
- **Our courses are focused - no "fluff" included.** We offer more than a "laundry list" approach to teaching. All lessons have clear objectives, are fundamental to core OOAD development and design practices, and are reinforced by hands-on labs and solid practical examples. Each lesson has performance driven objectives that ensure students will learn technologies and skills core to fundamental OO application and UML design – nothing more, nothing less.
- **Our materials are comprehensive, and current.** Our comprehensive manuals include not only a hard copy of the course presentation, but also detailed reference notes, pertinent diagrams and charts, current lists of suggested online resources and articles, and often technical tutorials or white papers geared to the topics at hand.
- **We set you up!** Hands-on courses also include our unique materials for each student, complete with our *LoadNGo Instant Classroom* course set up DVD, software, and a multitude of learning resources that complement the course. Run the course right off the DVD – minimal set up for your company – maximum results for your students.
- **We foster "Learning by Doing".** Progressive labs are designed in such a way that students get a firm grasp on fundamental skills while they work toward defending a complete application. All labs are take-home, and all solution code is presented in an easy to use self-study format for future use and review.
- **True content ownership gives us flexibility & quality above the rest.** These course materials are wholly-owned by our company and fully customizable - at little or no cost - to help you best meet your learning objectives. We have many dedicated experts available worldwide to instruct your team, and can provide services around the globe, either locally or virtually. We work closely with you to produce the most effective events and materials for your team, within your desired timeframe and budget.



- **We have to adhere to higher standards.** As a courseware provider, our content and hands-on lab materials are licensed internationally by dozens of firms, and are therefore subject to very stringent quality requirements. Not only will your organization benefit from our own technical team's technical expertise, but also the feedback of hundreds of students and trainers using these materials, worldwide, on a regular basis. This unique fact guarantees that our materials are not only robust and interesting, but also technically correct, current and of the highest quality and usability.
- **We bring years of practical, current experience into the classroom and content.** Our instructors and course authors are also skilled mentors, Java, J2EE, .Net, SOA, and web services developers, architects and security-oriented professionals. We believe that learning, using and maintaining solid software execution and delivery methods are as important as gaining sharp coding skills. Best Practices for software development and execution, beyond technical coding skills, are enforced throughout all of our courses and discussions. Our team brings this extensive experience into every classroom and engagement.
- **We're skills-centric.** Although our team has extensive experience using a variety of tools and solutions, our core content is "technology-centric". Our aim is to teach you the best skills and solutions out there – not to sell you software from any particular vendor.
- **We're Java & J2EE authors and industry speakers.** Our team was selected to write the online *J2EE, EJB, EJB CMP-CMR and Web Services Tutorial Series for IBM developerWorks®* (www.ibm.com) These are the same instructors who train our classes and author the courseware. Most of our trainers/consultants have also authored additional articles on web services, EJB< Struts, J2EE and advanced Java topics, and are recognized speakers and presenters on the industry technical seminar circuit. Our team is comprised on several successful published authors. Members of our team have written or contributed to: *Eclipse Kick Start, Mastering Eclipse; Professional Jakarta Struts; Using Java Tools for Extreme Programming; Mastering Resin; Mastering TomCat and others.*

