

Course 413: Overview of the Unified Modeling Language (UML) (4 days)

Course Description...

Since its inception in 1997, the Unified Modeling Language (UML) has rapidly become the standard for codifying the results of analysis and design. This course provides a review of all of the diagrams contained in the UML. The program is designed as an overview to assist managers, business analysts, and developers to read and understand the UML and create some of the basic diagrams. Approximately one half of the class is workshop. This course includes UML 2.0 diagrams and concepts.

Learning Objectives...

- Introduce all diagrams of the Unified Modeling Language 2.0.
- Be able to read and understand each of the UML diagrams.
- Know when and where to use each diagram to assist in defining quality systems.
- Understand the development processes associated with the UML.

Who should attend...

Applications programmers and designers, business analysts who must develop system level documentation.

Prerequisites...

Knowledge of design and analysis, and the object-oriented approach. A short review on the object-oriented approach is included.

See next page for a detailed course outline...



Course Outline...

Course Objective

Unit 1: Introduction to Modeling

Software Engineering

- System and software engineering
- Software architecture

Concept of Models

- Hierarchical decomposition
- Executable models
- Models
- Class Exercise: building a model
- Process
- What is a process
- Development processes that use UML

Unit 2: UML Overview

The UML Perspective

- Definition of the UML
- Where the UML came from
- Current Status of the UML specification

Review of Object-Oriented Technology

- Classes and objects
- Encapsulation
- Polymorphism
- Inheritance and aggregation

UML Characteristics and Notation

- Infrastructure and Meta-Object Facility
- Superstructure
- Diagram Interchange
- Object Constraint Language
- UML Profiles

Tool Usage

Class Exercise: Using the tool

Unit 3: The Models

Behavior and Requirements Models

- Defining the requirements
- Requirements and behavior notation

Analysis Models

- Why analyze information?
- Analysis notation

Design Models

- Creating the design
- Design notation



Implementation Models

- Physical implementation
- Implementation notation

Test Models

Unit 4: The Requirements Model

Defining the Problem

- The problem domain

Use Cases

- Definition
- Components
- Relationships between use cases
 - Includes
 - Extends
 - Generalization
- Use Case descriptions
- Class Exercise: creating the use case model

Packaging

- The package diagram
- Class Exercise: Packaging the use case model

Requirements

- Moving from use cases to requirements
- Class exercise: documenting requirements

Unit 5: Structure Diagrams

Class and Domain

- The domain model
- The object model
- The class diagram
- Class Exercise: Creating structure diagrams

Unit 6: Behavior Diagrams

Defining Behavior

- State machine diagrams
- Activity diagrams
- Timing diagrams
- Sequence diagrams
- Class Exercise: Creating Behavior Diagrams`

Unit 7: Interaction Diagrams

- Interaction Diagram
- Communication diagrams



Unit 8: Implementation Diagrams

Components

- Component diagrams
- Composite structure diagram
- Class exercise: Creating the Component Diagram

Deployments

- Deployment diagrams
- Class exercise: Creating the Deployment Diagram

Unit 9: Verifying and Refining

OCL and Constraints

- The Object Constraint Language
- The OCL standard library
- Tagged values

Using UML in the Test Model

- Transforming Use cases to test cases
- Applying the diagrams to test cases and procedures

Unit 10: The Bottom Line

Ideas to use

Where to go for more information

Please contact your ROI representative to discuss course tailoring!!!