



Advance OO Design and Component Based Development With UML

A Three Days Hands-On Course (RAT111)

Prerequisites: Attendance of Ratio's 5 day OOA/D using UML course, or substantial (6 months+) experience in OO design and a good understanding of UML.

Contents include:

Introduction to Components

Provides a basic overview of different types of components, and the notations used to represent them in UML.

- What is a component
- Component categorisations
- Logical and physical views of components
- Overview of component assembly
- UML is upside down
- Architectural reference model

Designing extensible components and frameworks

Gives students experience of designing components with the appropriate hooks to enable them to be customised, combined and extended.

- Designing the component interface (façade)
- Designing connection points
- Designing extension points
- Principles/patterns for extensible design

Assembling components

Shows students how to assemble components to meet a design objective.

- Adapting a component interface façade
- Connecting components to maintain integrity
- Extending a component to customise functionality
- Principles/patterns for extending components

Component Connection Technologies

Discusses component inter-connection technologies in overview and how they relate to the design content of this course.

- COM family
- CORBA
- Java Beans
- Enterprise Java Beans
- Inter-operability issues
- What they offer, what constraints they impose
- How they relate to this course



We Know the Object

Component based testing

Provision and use of appropriate testing harnesses during CBD.

- The testing process
- Providing the right test stub set

Architectural reference model for CBD

Discusses a five layer reference model to underpin large component based system development, and explains how it relates to advanced OO design principles.

- the reference model and its objectives
- rules of the model
- interface layer
- application layer
- domain layer
- infrastructure layer
- platform layer
- recursive application of the model to existing components
- summary and exercises

Advance object oriented design principles

Gives student practise at number of advanced object oriented design principles, both as stand-alone principles and in the context of component based development.

- Liskov Substitution Principle (LSP)
- Open Closed Principle (OCP)
- Common Closure Principle (CCP)
- Domain Separation Principle (DSP)
- Stable Dependencies Principles (SDP)
- Stable Abstractions Principle (SAP)
- Acyclic Dependency Principles (ADP)
- Dependency Inversion Principle (DIP)
- summary and exercises

Process related issues

Explains a process for component based development.

- identifying services from requirements use cases
- allocating services to components
- inter-component sequence diagrams (validating design)
- intra-component sequence diagrams (validating design)
- detailed design issues
- iterating the design

Documentation issues

Dicusses appropriate use of documentation to assist in re-use

- Encouraging re-use through appropriate documentation
- focus on the users view

