
OO and J2EE Best Practices Using UML, Java, Design Patterns, Rational Application Developer v6.x/v7.0, and Rational Software Architect v6.x/v7.0 for WebSphere

Course Code: WS401

Duration: 3.0 days

Objective

In this jam-packed 3.0 day course, you will gain practical and extensive Java/J2EE/object oriented design techniques by:

1. Learning OO/Java/J2EE best practices/design patterns developed by many OO mentors.
2. Studying through many real-life Java/J2EE case studies through over 35 design patterns.
3. Studying extensive set of Java/J2EE examples for Gang of Four design patterns, GRASP, Pattern Oriented Software Architecture (POSA), Patterns of Enterprise Application Architecture (PEAA), Sun J2EE Patterns, and IBM WebSphere Patterns.
4. Designing J2EE application using latest J2EE design patterns and IBM Rational Application Developer 6.x (IRAD6).
5. Learning Gang of Four design patterns support in Rational Rose 2003.x and Rational Software Architect 6.x (RSA6).
6. Applying Design Patterns with Various Open Sources Frameworks, J2SE, and J2EE Technologies.
7. Systematically analyzing, designing, implementing enterprise-scaled business, ebusiness, and engineering based software solutions.
8. Preview IBM Rational Application Developer/Rational Software Architect v7.x.

Topics

UML Review and UML to Java Mapping

- System Use Case
- Design Class diagram
- Communication Diagram
- Sequence Diagram
- State Machine Diagram
- Mapping Design Class Diagrams to Java Code
- Mapping Sequence/Communication Diagrams to Java Code

- Mapping State Machine Diagrams to Java Code

OO/J2EE Architecture Design Best Practices

- What is Design Pattern?
- Process of Applying Design Patterns
- Who Should Instantiate Objects
- Who Should Clean up Objects
- Who Should Map System Event to Business Services
- How to Design a Robust J2EE/Java Framework?
- What are Differences between Utility Classes and Frameworks?
- How to Provide Global Access Point in non J2EE and J2EE
- How to Model Concurrent Processes
- How to Ensure Single Instance is Created in non J2EE and J2EE
- How to Ensure Single Instance is Created and Thread Safe
- How to Model Service Request in non-J2EE and J2EE
- How to Distribute Service Requests to Multiple Servers
- How to Schedule Service Requests in non-J2EE and J2EE
- How to Handle Undo/Rollback Operations
- How to Provide Threading Service For A Transaction
- How to Make Multiple Views Consistent With A Single Data Model
- How to Manage Multiple Transactions or Service Requests
- How to Handle Multiple Undo/Rollback Service Requests
- How to Provide a Unified Interface From a Number of Subsystems
- How to Reduce Network Contention by Limiting Remote calls
- How to Transfer Aggregate Data across Different Architectural Layers
- How to Make Synchronous based Service Requests Scaleable
- How to Make Asynchronous based Service Requests Scaleable
- How to Enforce a Subclass Method Using A Fix Algorithm
- How to Materialize Relational Database Records To Objects
- How to Translate Workstation To Mainframe Data Format
- How to Defer Instantiation To Subclasses
- How to Design Asynchronous Requests by Object Locks?
- How to Restrict Access To Secure Data Sources
- How to Implement Cross Platform Solution
- How to Design Tree/Parent-Child Hierarchy based Solution
- How to Handle Objects with Complex Dynamic Behavior
- How to Simplify Distributed Calls and Free up Resources
- How to Design Transaction Framework
- How to Cache JNDI Lookup for JDBC, EJB, JMS Resource Managers

- How to Simplify Client Access to Session EJB
- How to Simplify Client Access to Message EJB
- How to Decouple Persistence Strategies
- How to Intercept Service Request/Response of J2EE Servlet/JSP
- How to Provide a Centralize Controller for all Related URL Request
- How to Provide a Helper Class for JSP to Perform Rendering Logic
- How to Provide a Helper Class for JSP to Show Computation Result
- How to Take Care of Duplicate Submission on Web Form Page

Applying Design Patterns with Various Open Sources Frameworks, J2SE, and J2EE Technologies

- Struts
- Web Service
- Swing
- EJB
- Servlet
- JDBC
- JSP
- JavaBean
- XML/XSL
- JDO
- JSF
- SDO

Prerequisite

Intermediate Java Programming or equivalent and 12 months of Java, J2EE (Servlet, JSP, JavaBean, EJB, XML), and UML experience. Familiarity with WebSphere Studio Application Developer 5.x (WSAD) or Rational Application Developer 6.x/7.x or Rational Software Architect 6.x/7.x is beneficial but not required.