Introduction to J2EE Design Patterns (4 Days)

Audience  
System architects, Java or OO developers, Project Managers and other professionals that will be designing or deploying web applications.

Course Abstract  
This course will utilize a combination of instructor-led discussions and interactive workshops to illustrate the use of patterns, best practices, design strategies and proven solutions for designing web applications that incorporate JSPs, Servlets, EJBs, JMS and JDBC components. This seminar will focus on the J2EE Patterns Catalog and design strategies that can be utilized to incorporate them. Each of the patterns for the Presentation, Business and Integration tiers will be extensively investigated with an in-depth discussion of the context, problem addressed, forces impacting each, solution provided and the related consequences.

Objectives  
Upon conclusion, each participant will have acquired these skills:

- Understand the concept of J2EE patterns and the Catalog.
- Illustrate the J2EE environment and the components affected by implementation of a J2EE pattern.
- Discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
- Depict various bad practices to be avoided when building Presentation tier components.
- Discuss the Business tier design patterns and their affect on: Session and Entity EJBs and the use of remote references.
- Depict various bad practices to be avoided when building Business tier components.
- Understand the variety of implemented bad practices related to the Business and Integration tiers.
- Examine the different patterns associated with the Integration tier.
- Highlight the evolution of patterns in the implementation of Struts and JavaServer Faces.
- Understand the role of J2EE Security
- Examine the optimization techniques available in the deployment of J2EE applications in a WebSphere environment.
- Depict the role of Cells, Clusters and Application Servers and impact of various patterns on different implementations.

Prerequisites  
Each student should have a basic understanding of application development and have been exposed to an object-oriented programming language.

Course Topics  
The following list represents the sections and topics discussed in this onsite instructor-led course offering:
J2EE Platform
- Architecture
  - Client tier
  - Web tier
  - Business tier
  - EIS tier
- J2EE Components & Containers
- Standard API Services (JMS, JDBC, JNDI, etc)
- Platform roles
- Business Object optimization

J2EE Design Pattern Overview
- What are Patterns
- Pattern benefits
- J2EE Pattern Catalog
  - Evolution
  - Presentation tier
  - Business tier
  - Integration tier

Presentation Tier Design Considerations
- Model-View-Controller
- Tier separation
- Design issues
  - Session Management
  - Client Session state
  - Client Session Security issues
  - Validation
- Bad Practices
  - Code control in multiple views
  - Expose Presentation-tier data to Business tier
  - Duplicate Form Submissions
  - Resetting bean properties via setProperty
  - Fat Controllers
  - Expose resources to direct Client access
  - Using Helpers as Scriptlets

Evolution of Presentation Patterns
- Struts Architecture
  - Overview
  - Stuts Pattern
  - Application Components
  - Tag usage
- JavaServer Faces Architecture
  - Overview
  - JSF Pattern
  - Application Components

Presentation Tier Patterns
- Interceptor Filter
  - Context
  - Problem
  - Solution & consequences
- Front Controller
  - Context
• Problem
  • Solution & consequences

• View Helper
  • Context
  • Problem
  • Solution & consequences

• Composite View
  • Context
  • Problem
  • Solution & consequences

• Dispatcher View
  • Context
  • Problem
  • Solution & consequences

• Service to Worker
  • Context
  • Problem
  • Solution & consequences

Business Tier Design Considerations
• Overview
  • Role of Session EJBs
    o Stateful vs Stateless
    o Use in Business-tier Facades
  • Store State in Business tier
  • Role of Entity EJBS
  • Business and Integration tier Bad Practices
    o Mapping object model to Entity EJB
    o Mapping relational model to Entity EJB
    o Use case to Session EJB mapping
    o EJB attribute exposure
    o Embedded Service lookup in Client tier
    o Read-only Entity EJB
    o Entity EJB as fine-grained objects
    o Exposing EJB exceptions to Client tier
    o Large result sets
    o EJB and long-lived transactions
    o Rebuilding conversation state with Stateless Session EJBs

Business Tier Patterns
• Business Delegate
  • Context
  • Problem
  • Solution & consequences

• View Object
  • Context
  • Problem
  • Solution & consequences

• Session Facade
  • Context
  • Problem
  • Solution & consequences

• Business Object
  • Context
  • Problem
  • Solution & consequences
• Composite Entity
  o Context
  o Problem
  o Solution & consequences
• Transfer Object
  o Context
  o Problem
  o Solution & consequences
• Transfer Object Assembler
  o Context
  o Problem
  o Solution & consequences
• Value Object Assembler
  o Context
  o Problem
  o Solution & consequences
• View List Handler
  o Context
  o Problem
  o Solution & consequences
• Service Locator
  o Context
  o Problem
  o Solution & consequences

Integration Tier Patterns
• Data abstraction via DAO implementations
  o JDO
  o SDO
  o EJB
  o JDBC
• Data Access Object
  o Context
  o Problem
  o Forces
  o Solution & consequences
• Service Activator
  o Context
  o Problem
  o Forces
  o Solution & consequences
• Domain Store
  o Context
  o Problem
  o Forces
  o Solution & consequences
• Web Service Broker
  o Context
  o Problem
  o Forces
  o Solution & consequences

J2EE Refactoring
• Presentation Tier
  o Controller
  o Synchronizer token
- Localize disparate logic
- Hide presentation
- Remove conversion from client

- **Business and Integration tier**
  - Wrap Entity with Session
  - Business Delegate
  - Merge Session EJBs
  - Reduce EJB communication

- **General Refactoring**
  - Separate Data access code
  - Connection Pool

---

**J2EE Security**
- WebSphere/J2EE Security model
- Authentication/Authorization
- Delegation and Trusting
- Network Deployment implementation
- LDAP vs Local OS registries
- Security Collaborators
- Security role mapping
- RunAsmode
- RunAsIdentity
- HTTP single sign-on

**WebSphere Performance Issues**
- Architecture
  - Cells
  - Clusters
  - Application Servers
- J2EE Application optimization
  - Resource Pooling
  - Sessions
  - Affinity issues
  - Failover
- WebSphere Performance Monitoring components
- Tivoli Performance Viewer and runtime performance
- PMI Service and infrastructure
  - Data collection services
  - Performance data classification
  - Enabling PMI
  - Instrumentation levels
- Monitoring real-time applications
- Viewing EJB/Web Container, JDBC pools, Threads, ORBs, etc