

**Course 470:**  
**Mastering the Java Enterprise Framework:**  
**JPA, EJB3, JSF and JAXWS**  
**(4 days)**

This in-depth workshop using Java Enterprise Edition 5.0 will provide developers with hands-on experience in designing and implementing enterprise applications using Java ServerFaces (JSF), annotations-driven Enterprise Java Beans (EJB 3.0), Java Persistence API (JPA) and the Java API for XML Web Services (JAX-WS).

**Who can benefit?**

This is an intermediate level Java programming course, designed for developers who wish to understand and develop using Java Enterprise Edition 5. The student should be an experienced J2EE / Java programmer, with practical development experience in Java.

**Course philosophy...**

Integrated development environment tools for JEE (such as Netbeans) provide graphical wizards to ease the development of Java EE applications. However, these tools hide quite a bit of detail, making it hard to troubleshoot already developed applications and making application development seem magical. Therefore, this course uses Eclipse (but without the EJB 3 wizards) and Ant to clearly delineate what exactly what is required of the developer and what the EJB container provides. Developers who take this class will obtain a clear understanding of the Java EE 5 application stack and of how to unit test components within that stack.

**Learning Objectives...**

- Map objects to relational databases using annotations or using configuration in a standards-compliant manner
- Configure container-managed and bean-managed transactions
- Inject dependencies using the Java EE dependency injection framework
- Implement a business tier with EJB 3.0 session beans and message-driven beans
- Expose EJBs through XML Web Services
- Implement an event-driven presentation tier using JSF
- Apply JEE security policies

**See next page for a detailed course outline...**



## Course Outline...

### Chapter 1: Architecting JEE Systems

- Evolution of JEE Systems, leading to Spring and Java EE
- EJB 2 issues: performance, maintainability and testability
- How Java EE addresses EJB 2 issues
- The JEE 3-Tier Architecture
- Core JEE Design Patterns
- Quick introduction to Eclipse, Ant and Tomcat
- **Exercise 1: Exploring the case study**
- Unit testing with JUnit
- **Exercise 2: Unit testing**

### Chapter 2: Java Persistence API

- Advantages of object-relational mapping
- Specifying entities
- Mapping rows to objects
- Changing JPA defaults using annotations
- Overriding annotations using orm.xml
- **Exercise 3: Specifying Entity Mapping**
- Managing entities
- Entity lifecycle
- Managed and detached entities
- Create, Read, Update, Delete operations
- Bean-managed transactions
- Entity relationships
- One to many, one to one, etc.
- **Check it out: One-to-many relationship, lazy fetching**
- Unidirectional and bidirectional relationships
- Java Persistence Query Language
- **Exercise 4: Managing entities and transactions**

### Chapter 3: Implementing the Business Tier with EJB 3.0

- Types of Enterprise Beans
- Performance considerations in EJB 3.0
- Session Beans
- Remote vs. Local Clients
- Method parameters
- Packaging and deploying a JEE application
- **Check it out: Writing, deploying a Stateless Session Bean**
- Testing a JEE application from JUnit
- Exceptions: runtime, checked and application exceptions
- **Exercise 5: Implementing and testing a Session Bean**
- Injecting resources
- Injecting EJBs
- Injecting EntityManager
- State management
- Stateful session beans
- Accessing a bean both remotely and locally
- **Exercise 6: Injecting into a Session Bean**
- Bean-managed Transactions
- Persistence Context
- Container Managed Transactions
- Application Exceptions
- Transaction Propagation



- Long-lived transactions: optimistic and pessimistic
- Session Façade Design Pattern
- How Session Facades map to users, use cases and transactions
- **Exercise 7: Implementing a Session Façade and Accessing it from a Rich Client**
- Building a Rich Client GUI with a EJB backend
- Message-driven Beans
- Timers
- Interceptors

## Chapter 4: Remote invocation of EJBs through Web Services

- What is JAX-WS
- Code-first or WSDL-first?
- Using JAX-WS to expose a EJB
- Deploying a web service
- **Check It Out: Coding and deploying a web service and getting its WSDL**
- Creating a web service client in Java
- **Check It out: Writing a web service client using JAX-WS**

## Chapter 5: Implementing the Presentation Tier with JSF

- Servlets, JSPs
- From Model 1 to Model 2
- State Management
- Session Façade and Business Delegate Design Pattern
- Architecture of JSF
- What does JSF provide?
- Organization of war and ear
- **Check It Out: Search Application Deployment**
- Managed Beans
- JSF Expression Language
- Injection into Managed Beans
- Stateful and Stateless beans
- Fast Track Access Design Pattern
- EntityManager, EntityManagerFactory
- UserTransaction
- **Check It Out: Search Bean**
- Navigation
- Views
- JSF Tag Libraries
- Built-in validators
- Form, input, command
- Select menus
- **Check it out: Search Application Views**
- Listeners
- Panel Grid
- Data Table
- Message Bundles
- **Exercise 8: Implementing a JSF Presentation Tier**

## Chapter 6: Java EE Security

- Declarative role-based security
- Securing web applications
- Declaring and mapping security roles
- Programmatic check of caller identity
- Declaring security requirements



- Specifying authentication mechanism
- **Exercise 9: Securing web application**

## **Chapter 7 [Optional]: Java EE Best Practices**

- Design best practices
- Testing Java EE components
- What to clean up
- Designing with Thread-safety of Java EE in mind

## **Chapter 8 [Optional]: Advanced JSF**

- Understanding the JSF lifecycle
- Short-circuiting lifecycle with exceptions
- Short-circuiting lifecycle with immediate
- Available converters
- Building a custom converter
- Validation
- Building a custom validator
- **Check it out: Custom conversion and validation**

*Please contact your ROI representative to discuss course customization!!!*