

## Course 470: Mastering the Java Enterprise Framework (4 days)

This in-depth workshop using Java Enterprise Edition 5.0 will teach developers how to design and implement enterprise applications using Java ServerFaces (JSF), annotations-driven Enterprise Java Beans (EJB 3.0), Java Persistence Architecture (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.

### 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 Architecture

- Limitations of JDBC
- How Hibernate improves on JDBC
- Entity and Key
- Persistence Factory and Entity Manager
- Mapping rows to objects
- Exercise 3: Implementing and testing a JPA DAO
- Create, Read, Update, Delete operations
- Java Persistence Query Language
- Entity Relationships: Multiplicity, Direction and Inheritance
- Exercise 4: Configuring relationships
- Container-managed Transactions
- Bean-managed transactions



- Multiple databases
- Try It Now: Setting up transactions

### **Chapter 3: Dependency Injection in Java EE**

- Looking up objects: Service Locator design pattern
- Issues with JNDI
- The Inversion of Control design pattern
- Dependency Injection in Java EE
- Limitations of Java EE dependency injection
- Injecting resources into servlet context listener
- Setting up a servlet context listener
- Exercise 5: Data access servlet
- Using JPA with other frameworks (Struts, Spring)

### **Chapter 4: Implementing the Business Tier with EJB 3.0**

- When to use Enterprise Beans
- Types of Enterprise Beans
- Performance considerations in EJB 3.0
- Session Beans
- Remote vs. Local Clients
- Method parameters
- Deploying a JEE application
- Testing a JEE application from JUnit
- Exercise 6: Writing, deploying and testing a Stateless Session Bean
- State management
- Message-driven Beans
- Try it now: Message-driven beans
- Life cycles of EJBs
- Using EJB3 with other frameworks (Struts, Hibernate)

### **Chapter 5: Remote invocation of EJBs through Web Services**

- Using JAX-WS to expose a EJB
- Deploying a web service
- Exercise 7: Writing and deploying an EJB as a WebService
- Creating a web service client in Java
- Exercise 8: Invoking a EJB as a web service from Java clients

### **Chapter 6: Implementing the Presentation Tier with JSF**

- Limitations of servlets and JSPs
- Model 2 Architecture
- Architecture of JSF applications
- Configuring JSF Navigation rules
- Lifecycle of a JSF page
- Tag libraries
- The component model
- Adding UI components to a page
- Standard converters
- Binding component values to external data sources
- Binding to backing bean properties
- Exercise 9: Developing JSP pages
- Custom error messages
- Custom localized text
- Try it now: Message bundles



- Writing bean properties
- Localization
- Implementing event listeners
- Custom validation
- Writing bean methods
- Exercise 10: Developing the model
- Transactions in web components
- Using JSF with other frameworks (Spring, Hibernate)

## **Chapter 7 [Optional]: Extending JSF**

- Creating a custom component
- Handling custom component events
- Registering a custom component
- Try it now: Custom component
- Creating a custom renderer
- Delegating rendering to a renderer
- Registering a custom renderer
- Try it now: Custom renderer
- Providing localized messages
- Try it now: Localization

## **Chapter 8: 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
- Http and Https
- Exercise 11: Securing web application
- Securing EJBs
- EJB security deployment descriptor elements
- Deploying secure EJBs

## **Chapter 9 [Optional]: Java EE Best Practices**

- Good OO and distributed design practices
- Naming conventions
- Testing Java EE components
- Measuring performance
- Designing with Thread-safety of Java EE in mind

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