



Open Source Live!

I HEAR and I FORGET. I SEE and I REMEMBER. I DO and I UNDERSTAND.

Developing Enterprise Java Applications with Lightweight Frameworks

Sacramento Course Details - \$300 OFF Super Special ends 3/30/07

Day 1: Track A Java Server Faces

This JSF course begins by explaining what JavaServer Faces is, and how it relates to Struts and other web frameworks currently on the market. You will learn about key JSF concepts, and some of the architectural principals behind the framework. You will then learn about IDEs that support JSF, as well as libraries that facilitate JSF development. You will also learn about the current market for off-the-shelf user interface components, such as grids, menus, toolbars, trees, and tabbed panes. We'll also examine the extension points within JSF, and how they can be leveraged to provide features such as security, alternate templating technology, and access to external resources.

- Introduction to JSF
- The Java web development landscape
- What is JSF?
- JSF fundamentals

Exploring the JSF landscape

- JSF implementations
- Third-party components
- Development tools
- Other frameworks and libraries

Day 1: Track B Hibernate 3

In this class you will learn how to use Hibernate to persist a domain model. We describe how to write the Hibernate ORM metadata for a domain model. You will develop data access objects (DAOs) using the Spring/Hibernate APIs.

Introduction to ORM with Hibernate

- Hibernate OR mapping
- Using the Hibernate API
- Executing queries
- Optimizing performance
- Using detached objects
- Handling concurrency
- Batch processing

Mapping objects to the DB with Hibernate

- Mapping classes
- Mapping relationships
- Transitive persistence
- Mapping inheritance

Getting Started

- Setting up your environment
- Structuring the application
- Developing views
- Navigation

Lab: Configuring a JSF application

- Get a JSF application to run
- Create a static login page
- Configure basic navigation

Creating Backing Beans and using Managed Beans

- The JSF Expression Language
- Managed Beans
- Backing Beans

Lab: Creating Backing Beans and using Managed Beans

- Configure application objects
- Write the AuthenticationBean backing bean
- Configure Authentication bean

Lab: Mapping objects to the DB with Hibernate

- Mapping classes and relationships
- Mapping user defined types
- Using transitive persistence

Developing DAOs with Hibernate

- The role of the data access layer
- Spring/Hibernate API
- Persisting, loading, querying and deleting objects
- Detaching and reattaching objects
- Hibernate queries

Register Now!

Day 2: Track A

Advanced Java Server Faces

Exploring the standard components

- Working with the standard components
- Using basic components and HtmlPanelGrid
- Using HtmlSelectItems
- Using HtmlDataTable

Internationalization, validators, and converters

- Internationalization

Day 2: Track B

Spring 2

In this class you will get a good understanding of the Spring framework including Spring's dependency injection and transaction management features. You will also, learn about the different design patterns for encapsulating the business logic and how Spring can be used with each pattern. In the labs you will use the Spring framework to make the business tier of an enterprise application transactional.

Introduction to the Spring framework

- Untangling crosscutting concerns
- Dependency injection

- The standard converters
- The standard validators

Lab: Working with the standard components

- Add JavaScript to the login page
- Develop the inbox page
- Write a backing bean

Developing with JSF

- Key APIs
- Building an application
- Best practices
- Security

Inside the JSF Architecture

- Using Phase Listeners and Pluggable Extensions

Lab: Using Phase Listeners and Pluggable Extensions

- Write an authorization phase listener
- Write a custom navigation handler

Writing custom components, validators, and converters

- Developing custom components
- Developing converters and validators

Wrap up & Future Directions

- Transaction management
- Database access with Spring
- Spring and testing

Business logic encapsulation options

- Avoiding the Golden Hammer Syndrome
- EJB Facade pattern
- POJO facade pattern
- Exposed Domain Model pattern

Transaction management with the Spring framework

- Transaction management as a crosscutting concern
- Declarative transaction management with Spring
- Annotations versus XML
- Custom aspects with Spring AOP and AspectJ

Lab: Implementing the Exposed Domain Model

- Configuring the OpenSessionInViewFilter
- Configuring transactions

Lab: Implementing the POJO façade pattern with the Spring framework

- Configuring transactions
- Detaching objects with Hibernate

Integrating Spring with the presentation tier

- JSF
- Struts
- Spring MVC

Register Now!

Day 3: Tack A

AJAX with JSF

This session is a perfect follow-on to two days Of JSF, where you will understand why AJAX is the great match for JSF. You will learn how to add AJAX functionality to JSF applications to deliver rich enterprise Internet applications without the tedious task of writing any JavaScript. You will comprehend various approaches to using AJAX with JSF. In the lab, you will build and test numerous AJAX/JSF examples using an open source framework called Ajax4jsf. You will explore this popular framework and learn how to effectively add AJAX capability seamlessly to JSF applications without any JavaScript coding. At the completion of this session, you will possess the knowledge and skills to build dynamic JSF application with AJAX.

Overview of AJAX

- Why use it?
- When to use it?
- Traditional Web applications
- AJAX Web applications (Web 2.0/Enterprise 2.0)
- AJAX technologies: JavaScript, XMLHttpRequest, DOM, CSS

Overview of AJAX Toolkits

- Sarissa
- Prototype
- Dojo Toolkit
- DWR
- Other

Introduction to Ajax4jsf

- What is Ajax4jsf?
- Architecture overview
- Ajax4jsf components

Lab: Implement Ajax4jsf Command Components

- a4j:support
- a4j:commandLink
- a4j:commandButton

Lab: Implement Ajax4jsf Container Components

Day 3: Track B

EJB 3 and JPA

One of the major changes in Java Enterprise Edition 5 (Java EE 5) has been in business components or the EJB3 tier. The new and improved EJB3 has gone lightweight with major changes in the Session and Message Driven beans and Entity bean replaced with JPA. In this session, we will focus on the JPA (Java Persistence API) entities, simplified development model, and review the changes in session and message driven beans. Since this is all about transferring know-how, be prepared for up to four hands-on labs that will include implementing entities for CRUD operations, along with implementing business logic in stateless, stateful and message driven beans. Time permitting; we will also look at design patterns and transaction management in EJB3. On completion of this session, you will have the knowledge and skills to implement lightweight EJB3 business component for Java EE. **Prerequisite:** Prior Java (JDK 1.5) and JSP programming experience along with previous J2EE knowledge and experience helpful.

Introduction to the EJB 3 architecture

- Agenda – what we'll will and won't cover
- Challenges with EJB 2.X
- What's new in EJB 3?
- Java Persistence API
- Changes in session bean
- Changes in message-driven bean
- The new, simplified development model
- Annotation
- Dependency injection
- Interceptors

JPA entities and the persistence framework

- The persistence unit, the entity manager, and the persistence context
- Specifying O/R mappings
- Defining entity relationships
- Defining queries using JPQL (Java Persistence Query Language)
- Mapping entity relationship hierarchies
- Using ID generators

Lab 1: Creating and testing JPA entities in a pure Java SE environment

- a4j:region
- a4j:page
- a4j:outputPanel

Lab: Implement More Ajax4jsf Components

- a4j:ajaxListener
- a4j:actionparam
- a4j:status
- a4j:loadBundle
- a4j:mediaOutput

Other Ajax4jsf Features

- Skinability

- Setup a JPA project
- Define a new persistence unit
- Create entities and map to database tables
- Create named queries using JPQL
- Create and execute a Java SE facade and test client

Session beans – stateless and stateful

- Defining session beans
- Annotation for SLSB and SFSB
- Using a session bean facade to manage JPA entities
- Exposing JPQL queries as facade methods
- Injecting resources
- Specifying transactional behavior

Lab 2: Creating and testing session beans

- Create a stateful session bean as a facade for the previously created entities
- Create a search stateless session bean
- Package and Deploy the session beans
- Create a Java client to perform persistent CRUD operations on the entities through the facade methods

Message driven beans

- Introduction to JMS and Messaging
- Defining MDB
- Using a MDB facade to manage JPA entities
- Injecting resources
- Specifying transactional behavior

Lab 3: Creating and testing a message driven bean

- Create a message driven bean as a facade for the previously created session bean
- Create a Java client to send message to message driven bean using P2P
- Package & Deploy the message driven bean
- Create a Java client to perform asynchronous operations on the entities through the facade methods

Design Patterns

- Java EE Design Patterns
- Java EE Anti-Patterns
- Review essential Java EE Design Patterns

Lab 4: Creating a Web application that interacts with session, message driven beans and entities

- Implement design patterns to tie the entity and session beans in the business tier
- Use a pre-created JSP front end to access the business tier
- Deploy the JSP and EJB3 application
- Execute the JSP form in a browser to perform persistent CRUD operations on the entities through the facade methods

Register Now!

Developing Enterprise Java Applications with Lightweight Frameworks

© 2007 Open Source Live™. All rights reserved.
Java, J2EE, JavaEE are registered trademarks of Sun Microsystems.