Introduction to Web Application Development Using JEE™, Frameworks, Web Services and AJAX

Duration: 5 Days *(Face-to-Face & Remote-Live)*, or 35 Hours *(On-Demand)*

Price: $2495 *(Face-to-Face & Remote-Live)*, or $1495 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more information.

Delivery Options: Attend face-to-face in the classroom, remote-live or on-demand training.

Students Will Learn

- Java Web application architecture
- Developing servlets and JSPs
- Consuming Web services
- Creating and deploying SOAP based and RESTful Web services
- Using JavaBeans in Web Applications
- Accessing databases with JDBC
- Importing and Utilizing JNDI Packages
- Utilizing the Spring dependency injection framework
- Managing database operations by using the Hibernate framework
- Building Web applications by using the Spring MVC framework
- Managing database transactions with Spring and Hibernate
- Effectively integrating Spring and Hibernate
- Integrating, testing and debugging AJAX functionality on both the client and server sides
- Working with the JSON data format

Course Description

This course provides students hands on experience with cutting edge Java Enterprise (Java EE) technologies, creating dynamic web and enterprise applications that utilize several Java frameworks and technologies including JSP's and Servlets, Java Persistence API (JPA), JNDI, JDBC, AJAX, Web Services, Spring and Hibernate. The goal is to enable students to exploit the Java EE platform and accompanying frameworks to facilitate the development of distributed, web-enabled applications.

Students will architectural design issues as well as specific coding models for a variety of Java EE components. By working with several Java frameworks in hands on labs, students will build applications that incorporate many of the patterns commonly used in these and other Java frameworks. Upon completion of the course, students should be able to learn and effectively utilize frameworks appropriate for their application environment.

Starting with Java Server Pages and Servlets, the course then introduces some of the most widely used frameworks to provide concrete illustrations of the services available. Since coding and deployment files are standardized by the Java EE specifications, students may readily apply the skills learned in this class to write code for any compliant server, including Apache Tomcat, JBoss, WebSphere, Oracle, WebLogic and many others.

Students will learn how to utilize ANT, a flexible and powerful XML-based build utility, to compile, deploy and execute stand-alone and enterprise Java applications. They will also use ANT to execute standalone client applications that communicate with Java EE applications. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students who wish to focus more in-depth on Spring and Hibernate should attend the course Web Application Development Using Spring, Hibernate and JPA rather than this course.

Course Prerequisites

Java SE programming experience and an understanding of object-oriented design principles. Fundamental knowledge of XML, HTML, and JavaScript is helpful but not required. HOTT's course Java Programming or equivalent knowledge provides a solid foundation.

Course Overview

Introduction to Java EE Web Applications

- Server-Side Application Development using Java EE
- The Role of Java Frameworks, Components and Services
- Using Web-Based Components in Application Design
- Structure of Java EE Web Components
- Deploying Web Applications
- Java EE Web Container Services

Using ANT

- Understanding the ANT Build File
- Designing ANT Targets
- Using ANT Standard Tasks
- Using ANT Properties
- Compiling and Executing Java Applications
- Building WAR, EAR and JAR Deployment Files

Introduction to Servlets

- Servlet Architecture and Advantages
- The Role of Servlets in Web Application Design
- Servlet Runtime Environment
- Servlet Lifecycle

Developing Servlets

- Servlet Classes and Interfaces
- Working with Request and Response Objects
- Processing `GET` and `POST` Requests from Web Clients
- Retrieving Parameters from HTML Client Forms
- Generating Dynamic HTML Responses
- Initializing Servlets
- Destroying and Freeing Resources in Servlets
- Controlling Single and Multi-Threading in a Servlet
- Deploying Servlets to a Web

Application Server
- Building the WAR file
- The `web.xml` Descriptor File

Developing Java Server Pages (JSPs)

- Understanding JSP/Servlet Translation
- Elements of JSP Syntax
- JSP Page Directives
- JSP Declarations
- Displaying JSP Expressions
- Writing Scriptlets
- Deploying JSPs
- Using JavaBeans in JSPs
- JavaBean Architecture
- Creating JavaBeans
- Using JavaBeans in JSP Pages
- XML-format JSP Documents

Deploying and Using Tag Libraries

- Motivation for Tag Libraries
- Implementing the Model-View-Controller Pattern
- JSP Built-In Actions
- The JSP Standard Tag Library (JSTL)
- Deploying and Using Tag Libraries

Writing Tag Handlers

- Analyzing Tag Library Descriptor Files
- Creating Tag Library Descriptor Tags
- Using Tag Handler Methods to Generate Dynamic Web Content
- Handling Tag Attributes and Attribute Validation
- Processing Tag Body Content
- Working with the TagExtraInfo Class
- Developing and Deploying Tag Files

Accessing Databases with JDBC

- Understanding the JDBC Connectivity Model
- Accessing Data Sources through JNDI
- Connecting to a Database
- Executing SQL Queries and Updates
- Processing Result Sets
- Using Scrollable and Sensitive Result Sets
- Working with `ResultSetMetaData` Classes
- Utilizing Parameterized Statements
- Calling Stored Procedures
- Handling `SQLExceptions`
- Controlling Transactions
- Using Batch Updates

Java Naming and Directory Interface (JNDI)

- Role of JNDI in the Java EE Architecture
- JNDI Service Providers
- Importing and Utilizing JNDI Packages
- Binding Objects with JNDI
- Looking up Objects with JNDI
- Using the Environment Naming Context (ENC)
- Declaring Resource References

Hibernate Overview

- Need for Hibernate
- Hibernate and ORM (Object-Relation Mapping)
- POJOs (Plain Old Java Objects) and the Data Layer
- Hibernate Mapping
- Hibernate Over Entity Beans

Hibernate Programming Fundamentals

- Modeling Complex Mappings
- Hibernate and Native SQL
- HQL (Hibernate Query Language)
    - `Where`

Hibernate Power Programming

- Hibernate Annotations and JPA
- Issues with Adding Hibernate to Existing Systems
- Developing POJOs in Existing Systems

    - `Order By`
    - `Group By`
- Hibernate Aggregate functions
    - `Avg`
    - `Min`
    - `Max`
- HQL Associations and Joins
- Hibernate Subqueries
- Hibernate Configuration Files
- Full CRUD Application (Create, Retrieve, Update, and Delete)
- Advanced HQL Techniques
- HQL Result Transformers
- Using the Criteria API for Complex Queries
- Transaction and Concurrency Issues
- Identifying and Removing Performance Bottlenecks
- Lazy Loading for Performance
- Inheritance and Polymorphism in the Persistence Layer
- Hibernate Best Practices

Introduction to the Spring Application Framework

- Spring Capabilities and Modules
- The Role of POJOs
- Lightweight IOC (Inversion-Of-Control)
- Dependency Injection
- The Factory Pattern and the Spring Container
- Spring XML Configuration Files
- Spring Persistence Support
- Data Access Framework and Data Access Objects
- AOP (Aspect-Oriented Programming)

Spring Framework Core Components

- Spring Core
- Using POJOs (Plain Old Java Objects)
- Spring Bean Factory
- Singleton and Prototype Beans
- Setter Injection and Constructor Injection
- Order of Instantiation
    - Lazy vs. Eager
- Controlling Spring's XML Configuration Files

Adding Spring to Existing Applications

- Issues Adding Spring to Existing Applications
- AOP and Transactions
- Spring Valuators
- Spring Interceptors
- Spring MVC
- Integrating Spring with Hibernate
- Possible Conflicts with Existing Systems
- AOP vs. Annotations

Introduction to AJAX

- AJAX Architecture and Capabilities
    - Client Side
    - Server Side
- Circumventing the Page Reload Paradigm
- CSS, HTML and AJAX
- JavaScript and DOM
- `XMLHttpRequest` Object
    - `readyState` and `responseXML` Properties
- Making AJAX Asynchronous Calls
- AJAX Function Calling Conventions
- Response Handling with JavaScript
- Browser Compatibility Issues
- Server-Side and Client-Side AJAX

Java and AJAX

- Callback Functions/Methods
- Built-In Objects
- Parsing AJAX Responses
- XML and JSON Response Formats
- Servlet Code to Handle AJAX Requests
- Server Side Persistence
- AJAX Patterns and Best Practices

Adding AJAX to Existing Web Applications

- Issues Adding AJAX to Existing Apps
- Dealing with Asynchronous Responses
- Cross Browser Libraries and Frameworks
- Working with AJAX Toolkits
    - Dojo Toolkit

- Limitations of AJAX
- Debugging AJAX
    - Prototype
    - DWR (Direct Web Remoting)
    - Google Web Toolkit
- AJAX and REST Design Compatibility Issues
- Security Issues

Web Services on the Server Side

- Overview of Web Services
- Advantages of Web Services
- Creating a Web Service
- Deploying a Web Services
- Requirements for a JAX-WS Web Service Endpoint Implementation Class
- The Web Service Endpoint Implementation Class
- Working with WSDL Files
- Web Service Interoperability

Web Services on the Client Side

- Consuming a Web Service
- Client Side Artifacts
- JAX-WS Clients
- Java/WSDL Mapping
- RESTful Web Services
- RESTful Web Services in Java

Hands On Technology Transfer
The Best Way to Transfer Technology Skills

1 Village Square, Suite 8
14 Fletcher Street
Chelmsford, MA 01824