Developing Applications for the Java EE 7 Platform Ed 1

JAVA

DURATION	MODULES	COURSE CODE
5 Days	**12 Lectures**	**—**

Course Overview

The Developing Applications for the Java EE 7 Platform training teaches you how to build and deploy enterprise applications that comply with Java Platform, Enterprise Edition 7. The technologies presented in this course include annotations, Enterprise JavaBeans (EJB), Java Persistence API (JPA), Java Transaction API (JTA), Servlets, JavaServer Pages(JSPs), JavaServer Faces (JSF), Contexts and Dependency Injection (CDI), JAX-RS RESTful and SOAP Web Services, the Java API for WebSocket, Java Message Service API (JMS), Bean Validation, Batch API, Timer services, and Java EE Concurrency.

What You Will Learn

Course Introduction

- Course Objectives
- Audience
- Class Introductions
- Course Environment
- Course Structure
- Course Practices
- Course Appendices
- Course Schedule
- Summary

Introduction to Java EE

- Objectives
- Requirements of Enterprise Applications
- Separation of Business Logic from Platform Services
- Structure and Purpose of Java EE 7 Server, Containers, and APIs
- EJB Lite and EJB Full Containers
- Evolution of Web Design
- MVC (Model View Controller)
- Java EE Web Container Components: Servlets

- Java EE Web Container Components: JSPs
- Java EE Web Container Components: JSFs
- Java EE Web Container Components: REST Services
- Java EE Web Container Components: Web Sockets
- Java EE 7 Web Services
- Java EE 7 Business Logic Handling Technologies
- Maintaining Application State
- Session EJB Types
- Message-Driven EJB
- Assembling Application Components with CDIs
- JSF Managed Beans, CDI Beans, EJBs
- Request Scope
- Session Scope
- Application Scope
- View Scope
- Conversation Scope
- Dependent Scope
- Injecting Beans
- Java EE Packaging and Deployment
- Annotations or Deployment Descriptors
- Annotations with Deployment Descriptors
- Java Naming Directory Interface Objects
- Container-Managed Injections
- JNDI Lookups
- Summary
- Practices

Managing Persistence by Using JPA Entities

- Objectives
- Java Persistence API
- JPA Entities: Basics
- Persistent Field Versus Persistent Property
- Using Access Annotation
- Converters
- Generated Keys
- JPA Lifecycle Callback Methods
- Validating Entities
- Using Bean Validation Constraints
- Container Managed Persistence
- Locally Managed Persistence
- Entity Manager Operations
- Locking and @Version
- Changing Locking Mode
- Java Persistence Query Language (JPQL)
- Using JPQL with non-Entity classes

- Executing JPQL Statements
- Summary
- Practice Overview

Implementing Business Logic by Using EJBs

- Objectives
- EJBs and EJB Container
- Enterprise JavaBean Types
- Session EJBs
- Accessing Session Beans
- Stateless Session Bean Life Cycle
- Stateful Session Bean Life Cycle
- Singleton Session Bean Life Cycle
- Asynchronous EJB operations
- Java Transaction API
- Programmatic Transactions (BMT)
- Declarative Transactions (CMT)
- Demarcate Transactional Attributes
- Transaction Scoped Beans
- Timers
- Calendar-Based Timer Expressions
- Define Programmatic Timers
- Define Automatic Timers
- Manage Timers
- Define Interceptors
- Types of Interceptors
- Apply Interceptors
- Summary
- Practice Overview

Using Java Message Service API

- Objectives
- Java Message Service (JMS) API
- JMS Destination Types
- JMS 2.0 API
- JMS Context
- Java SE Message Producer
- Java SE Message Consumer
- Java SE Asynchronous Producers and Consumers
- JMS Session Modes and Message Acknowledgments
- Handle JMS Messages
- Java EE Message Producer
- Java EE Message Consumer
- Topics Shared/Unshared Subscriptions
- Queue Message Browser

- Message-Driven Bean (MDB)
- MDB Life Cycle
- JMS and Transactions
- Handle Errors with Transactions
- Summary
- Practice Overview

Implementing SOAP Services by Using JAX-WS

- Objectives
- WebServices and SOAP
- Web Service Interaction Patterns
- Web Service Interface
- XML Schema Definition
- WSDL Schemas and Namespaces
- WSDL Messages, PortTypes, and Operations
- WSDL Bindings and Services
- Top-down versus Bottom-up approach
- Map Java Interface to WSDL
- JAX-WS Implementation
- Create JAVA JAX-WS Client
- Invoke SOAP Service from Java SE Client
- Invoke SOAP Service from Java EE Component
- Summary
- Practice Overview

Creating Java Web Applications by Using Servlets

- Objectives
- HTTP Protocol Basics: Sending Requests
- HTTP Protocol Basics: Getting Responses
- Create Servlet
- Override Servlet Request Handling Operations
- Provide Request Handling Logic
- Retrive Request Headers
- Retrieve Parameters
- Use Cookies
- Produce Different Content Types
- Manage Servlet Life Cycle with Container Callbacks
- CDI Beans
- HTTP Session Tracking
- Web Container Life Cycle Events
- Request Dispatcher
- Servlet Filters
- Asynchronous Servlets
- Nonblocking I/O
- Handle Errors

- Summary
- Practice Overview

Creating Java Web Applications by Using JSPs

- Objectives
- Create Java Server Page
- Java Server Page Syntax
- Java Server Page XML Syntax
- Expression Language
- Expression Language Operators
- JSP Scopes and Implicit Objects
- Use CDI Beans in JSPs
- Standard Tag Library (JSTL)
- Create JSP Error Handlers
- Summary
- Practice Overview

Implementing REST Services using JAX-RS API

- Objectives
- REST Service Conventions and Resources
- REST Communication Model
- Implementing REST Services using JAX-RS API
- Mapping Resources to URI Paths
- Mapping REST Resource Operations
- Handling Different Media Types
- Passing Parameters
- Validating Values
- Handling Web Service Errors
- Asynchronous REST Services
- Asynchronous EJB and REST Services
- Invoking REST Service from JavaScript Client
- Invoking REST Service from Java Client
- Invoking REST Service from Asynchronous Java Client
- Summary
- Practice

Creating Java Applications with WebSockets

- Objectives
- WebSockets Network Protocol
- WebSocket Life Cycle
- Defining WebSocket Endpoints
- Using PathParam Annotation
- Using WebSocket Session
- Using RemoteEndpoint Objects

- Encode and Decode Messages
- Handle WebSocket Messages
- Handle WebSocket Errors
- Encoding and Decoding WebSocket Messages
- Implementing WebSocket Message Encoder
- Implementing WebSocket Message Decoder
- Creating JSON Messages
- Parsing JSON Messages
- Invoking WebSocket Server from a JavaScript Client
- Invoking WebSocket Server from a Java Client
- Summary
- Practice

Developing Web Applications Using JavaServer Faces

- Objectives
- JavaServer Faces Concepts
- Faces Servlet Registration
- JSF Configuration
- JSF Facelet Structure
- JSF Request-Response Lifecycle
- JSF Libraries
- JSF HTML Library UIComponents
- JSF HTML Passthrough
- Using Validators and Converters
- JSF Templates
- Describe JSF Navigation
- Configuring Navigation Rules
- Using Faces Flows
- Action and ActionListener Attributes
- Value and Binding Attributes
- Using immediate attribute
- Using FacesContext Object
- JSF Localization
- Displaying Messages
- Producing Messages From CDI Beans
- Using Managed Properties
- Adding AJAX code to Facelets
- Extended JSF Frameworks and Component Libraries
- Summary
- Practice

Securing Java EE Applications

- Objectives
- JAAS Security Concepts
- JAAS Configuration

- Request Authentication and Authorization
- Login Module Configuration
- Programmatic Authentication
- Declare Application Roles
- Define Security Constraints
- Java EE Programmatic Security
- Web Service Security
- WS-Security
- Summary
- Practice
- A Java Logging
- Objectives A-2
- Java Logging Frameworks A-3
- Using Java Logging API A-4
- Logging Method Categories A-5
- Guarded Logging A-7
- Log Writing Handling A-8
- Logging Configuration A-10
- Application Sever Logging Configuration A-11
- Configuring the WebLogic Logging Service A-12
- Viewing WebLogic Server Logs A-13
- B CDI Beans
- Objectives B-2
- Using Named Qualifiers B-3
- Using Custom Qualifiers B-4
- Using Alternative Qualifiers B-5
- Producer and Disposer Methods B-6
- Interceptors B-7
- CDI Events B-8
- Stereotypes B-9
- C BeanValidation and JPA API
- Objectives C-2
- Custom BeanValidation Constraints C-3
- Entity Relationship Types C-4
- Mapping Entity Relationships C-6
- Entity Relationship Mapping Properties C-7
- Mapping Embeddable Classes C-8
- Mapping an Entity to Multiple Tables C-9
- Composite Primary Key C-11
- Using Inheritance with Entities C-13
- Using Unmapped Superclass C-14
- Using Mapped Superclass C-15
- Entity Inheritance Mapping Strategies C-16
- D Batch and Concurrency APIs
- Objectives D-2
- Concurrency D-3

- Executor Service D-5
- Managed Task Listener D-6
- Batch Processing: Overview D-7
- Job Specification Language (JSL) XML Structure D-8
- Batch API Structure D-12
- Describe Job Using JSL XML Document D-14
- Run Batch Job D-15
- E JAXB API
- Objectives E-2
- JAXB API E-3
- Bind Java Classes to XML Schema E-4
- Read and Write XML with JAXB E-5
- JAXB Annotations Part I E-6
- JAXB Annotations Part II E-7
- F "Pre-CDI" Servlet Examples
- Objectives F-2
- Using Request Application Attributes Without CDI F-3
- Using HttpSession Attributes Without CDI F-4