Developing Applications for the Java EE 6 Platform (FJ-310-EE6)

Duration: 5 Days

What you will learn

The Developing Applications for the Java(TM) EE Platform course provides students with the knowledge to build and deploy enterprise applications that comply with Java(TM) Platform, Enterprise Edition 6 technology standards. The enterprise components presented in this course include Enterprise JavaBeans(TM) (EJB(TM)) technology, the Java Persistence API, servlets, and JavaServer Pages(TM) (JSP(TM)) technology, JavaServer Faces(TM) (JSF(TM)), RESTful and SOAP web services, and the Java technology clients that use them.
Students gain hands-on experience through labs that build an end-to-end, distributed business application. The labs explore session EJB components, which implement the Session Facade pattern and provide a front-end to entity components using the Java Persistence API. The labs also explore message-driven EJB components, which act as Java Message Service (JMS) consumers. Students create user interfaces using servlets, JSP technology (JSP pages), and JavaServer Faces (JSF). Basic web services using SOAP and RESTful techniques will be created. Students learn how to assemble an application and how to deploy an application into an application server (Java EE platform runtime environment).Students perform the course lab exercises using NetBeans(TM)Integrated Development Environment (IDE).
Students who can benefit from this course
* Sun(TM) Certified Java technology programmers who want to develop enterprise applications that conform to the Java EE platform standards.
* Students with Java Programming experience interested in broad overview of the Java EE platform.
* Students planning to pursue one or more of the Enterprise Java EE6 certification exams.

This course counts towards the Hands-on course requirement for the Java EE 5 Enterprise Architect Certification. Only instructor-led inclass or instructor-led online formats of this course will meet the Certification Hands-on Requirement. Self Study CD-Rom and Knowledge Center courses DO NOT meet the Hands-on Requirement.

Prerequisites

Required Prerequisites
Experience with the Java programming language

Familiarity with object serialization

Familiarity with relational database theory and the basics of structured query language (SQL)

Familiarity with the use of an IDE

Suggested Prerequisites
Java Programming Language, Java SE 6

Course Objectives

Describe the application model for the Java EE platform and the context for the model
Select the correct Java EE Profile for a given application
Develop and run an EJB technology application
Develop basic Java Persistence API entity classes to enable database access
Develop a web-based user interface using Servlets, JSPs, and JSF
Develop simple web services for the Java EE platform

Course Topics

Survey of Java EE Technologies
Describe the different Java platforms and versions
Describe the needs of enterprise applications
Introduce the Java EE APIs and services
Certifications Paths
Introducing Applications Servers
Enterprise Modules

Enterprise Application Architecture
Design Patterns
Model View Controller
Synchronous and Asynchronous communication
Network Topologies and Clustering
Layering (client,presentation,service,integration,persistence)

Web Technology Overview
Describe the role of web components in a Java EE application
Define the HTTP request-response model
Compare Java servlets, JSP, and JSF
Brief introduction to technologies not covered in detail

Developing Servlets
Describe the servlet API
Servlet configuration through annotations and deployment descriptors
Use the request and response APIs
Servlets as controllers

Developing With JavaServer Pages Technology
Evaluate the role of JSP technology as a presentation mechanism
Author JSP pages
Process data received from servlets in a JSP page
Brief introduction to the JSTL and EL

JavaServer Faces
The JSF model explained
Adding JSF support to web applications
Using the JSF tag libraries
Configuring JSF page navigation
JSF Managed beans
JSF Conversion, Validation, and Error Handling

EJB Overview
EJB types: Session Beans
EJB types:Message Driven beans
Java Persistence API as a replacement for Entity EJBs
Describe the role of EJBs in a Java EE application
EJB lite

Implementing EJB 3.0 Session Beans
Compare stateless and stateful behavior
Describe the operational characteristics of a stateless session bean
Describe the operational characteristics of a stateful session bean
Describe the operational characteristics of a singleton session bean
Create session beans
Package and deploy session beans
Create session bean clients

The Java Persistence API
The role of the Java Persistence API in a Java EE application
Object Relational Mapping
Entity class creation
Using the EntityManager API
The life cycle and operational characteristics of Entity components
Persistent Units and Packaging

Implementing a Transaction Policy
Describe transaction semantics
Compare programmatic and declarative transaction scoping
Use the Java Transaction API (JTA) to scope transactions programmatically
Implement a container-managed transaction policy
Support optimistic locking with the versioning of entity components
Support pessimistic locking of entity components
Using transactions with the web profile

Developing Asynchronous Java EE Applications and Messaging
The need for asynchronous execution
JMS technology introduction
List the capabilities and limitations of Java EE components as messaging producers and consumers
JMS and transactions
JMS administration

Developing Message-Driven Beans
Describe the properties and life cycle of message-driven beans
Create a JMS message-driven bean

Web Service Model
Describe the role of web services
Web service models
List the specifications used to make web services platform independent
Describe the Java APIs used for XML processing and web services

Implementing Java EE Web Services with JAX-WS and JAX-RS
Describe endpoints supported by the Java EE 6 platform

Developing Web Services with Java
Creating Web Service Clients with Java

Implementing a Security Policy
Exploit container-managed security
Define user roles and responsibilities
Create a role-based security policy
Use the security API
Configure authentication in the web tier