Architect Enterprise Applications with Java EE

Duration: 5 Days

What you will learn

This Architect Enterprise Applications with Java EE training teaches you how to develop robust architectures for enterprise Java applications using the Java Platform, Enterprise Edition (Java EE) technology. Develop effective decision-making skills through using non-functional qualities (such as scalability and flexibility), Java EE technology blueprints and design patterns.

Learn To:

Define the enterprise architect’s roles, responsibilities and deliverables.
Identify non-functional requirements (NFRs) and describe common problems and solutions.
Translate business requirements into an architecture.
Weigh choices in architecting the client, web, business, integration and data tiers.
Apply various evaluation criteria to choosing architectural elements and patterns, tools, servers and frameworks.
Prepare for the Oracle Certified Enterprise Architect exam.

Benefits to You

By taking this course, you'll walk away with a deeper understanding of the technical context of the Java EE and relevant technologies. You'll be familiar with strategies needed to create application blueprints that work well when implementing Java EE technologies. Furthermore, expert Oracle University instructors will help you gain insight into the role of the enterprise architect as you learn how to use Java EE technologies in n-tier enterprise systems.

Ideal for Developers

This course will is ideal for developers who are responsible for the overall software architecture and design of Java EE technology-based enterprise software systems. Existing architects will explore using Java EE technologies to improve quality of service in enterprise systems.

Live Virtual Class Format

A Live Virtual Class (LVC) is exclusively for registered students; unregistered individuals may not view an LVC at any time. Registered students must view the class from the country listed in the registration form. Unauthorized recording, copying, or transmission of LVC content may not be made.

Audience

Architect

Developer
J2EE Developer
Java EE Developers

Related Training

Required Prerequisites

Describe distributed computing and communication concepts

Describe, in outline form, all Java EE technologies, including Enterprise JavaBeans, servlets, JavaServer Pages, and JavaServer Faces

Perform analysis and design of object-oriented software systems

Use a notation, such as the UML, for modeling object-oriented systems

Object-Oriented Analysis and Design Using UML

Developing Applications for the Java EE 6 Platform

Suggested Prerequisites

Java Design Patterns

Java EE 6: Develop Business Components with JMS & EJBs

Java EE 6: Develop Web Components with Servlets & JSPs

Course Objectives

Derive software systems using techniques outlined in the Java EE Blueprint and solutions defined in the Java EE Patterns

Address quality-of-service requirements in a cost-effective manner using engineering trade-off techniques

Describe the role of the architect and the products an architect delivers

List and describe typical problems associated with large-scale enterprise systems

Make good use of Java EE component technologies to solve typical problems in system architecture

Course Topics

Introducing Enterprise Architecture

What is Enterprise Architecture?
An Architect’s Roles and Responsibilities

Introducing Fundamental Architectural Concepts

Distinguish between architecture and design
Architectural Patterns
Architectural Deliverable Artifacts
What is an Enterprise Architecture Framework
4 + 1 View Model
Architectural Modeling Using UML
Architecture Workflow
What is an Enterprise Architecture Framework

Developing a Security Architecture
Analyzing the Impact of Security in Distributed Computing
Examining Security in the Java EE Technology
Understanding Web Services Security

Understanding Non-Functional Requirements
Examining Non-Functional Requirements (NFRs)
Common Practices for Improving Qualities
Prioritizing Quality-of-Service (QoS) Requirements
Inspecting QoS Requirements for Trade-offs

Defining Common Problems and Solutions: Risk Factors and System Flexibility
Identifying Risk Factors
Designing a Flexible Object Model

Defining Common Problems and Solutions: Network, Transaction and Capacity Planning
Describing Network Communication Guidelines
Justifying the Use of Transactions
Planning System Capacity

Java EE 6 Overview
Java EE 6 Goals
Java EE Containers
Classic Java EE 5 Architecture
Impact of Java EE 6 on Architecture

Developing an Architecture for the Client Tier
Client Tier Development Roles
Information Architecture Client Concerns
Selecting User Interface Devices and Technologies
Discovering Reusability in the Client Tier
Deployment Strategies for the User Interface
Security Concerns in the Client Tier
Testing

Developing an Architecture for the Web Tier
Responsibilities of the Web Tier
Separation of Concerns
Comparing Web Tier Frameworks
Providing Security in the Web Tier
Scaling the Web Tier

Developing an Architecture for the Business Tier

Business Tier Technologies
Architecting the Domain Model
Development Best Practices

Developing an Architecture for the Integration and Resource Tiers
Examining Enterprise Information System Integration
Reviewing Java Integration Technologies
Applying Integration Patterns
Examining Service-Oriented Architecture (SOA)

Evaluating the Software Architecture
Evaluating Software Architectures
Evaluating Java EE Technologies
Creating System Prototypes
Selecting Servers and Frameworks