1 Introduction

Objectives

After completing this lesson, you should be able to:

- Identify the course objectives
- Discuss the course agenda
- Discuss the contents covered in the course

Course Objectives

After completing this course, you should be able to:

- Describe the Java EE 5 standards and their APIs
- Design Java EE applications by using the Model-View-Controller (MVC) architecture and Session Facade pattern
- Build the business tier by using the Enterprise JavaBeans 3.0 (EJB) components, Java Persistence API (JPA) entities, and Web services
- Build the Web tier by using the JavaServer Faces (JSF) framework
- Implement asynchronous messaging by using Message-Driven Beans (MDB)
- Secure the Java EE application with Java Authentication and Authorization Service (JAAS)

Course Agenda

Topics for the first day

- Lesson 1: Introduction
- Lesson 2: Fundamentals of Java EE Technology
- Lesson 3: Designing Java EE Applications
- Lesson 4: Developing a Web Application Using Servlets

Course Agenda

Topics for the second day

- Lesson 5: Developing a Web Application Using JavaServer Pages (JSP)
- Lesson 6: Accessing Resources with JNDI and Dependency Injection
- Lesson 7: Developing the Business Logic with Session Beans

Course Agenda

Topics for the third day

- Lesson 8: Developing the Persistence Layer with JPA Entities
- Lesson 9: Manipulating JPA Entities with the `EntityManager` API
- Lesson 10: Developing the Business Logic with Web Services
- Lesson 11: Developing the Web Interface Using JavaServer Faces

Course Agenda

Topics for the fourth day

- Lesson 12: Planning Navigation and Page Flow
- Lesson 13: Handling Application Events
- Lesson 14: Asynchronous Communication with Message-Driven Beans

- Describe the Java Platform, Enterprise Edition (Java EE) platform.
- Define the components of a Java EE application.
- Identify the deployment options for a Java EE application.
- List the security options available in Java EE applications.

Designing Java EE Applications

- Identify the Java EE design patterns and their purpose.
- Describe the Model-View-Controller (MVC) architecture.
- Define the purpose and role of Struts.
- Define the purpose and role of JSF.
- Describe some important concepts and terminology used in Oracle WebLogic Server.

Developing a Web Application Using Servlets

- Define the role of servlets in a Java EE application.
- Describe the servlet life cycle.
- Describe the request and response architecture.
- Implement the HTTP servlet methods.
- List the Java EE servlet mapping techniques.
- Handle errors in a servlet.
- Create and run a servlet in JDeveloper.

Developing a Web Application Using JavaServer Pages

- Describe the relationship between JSP and servlets.
- List the implicit objects on JSP pages.
- Describe the semantics of JSP tags.
- Create a JSP segment.
- Explain the use of JSP tag files.
- Describe custom tags and expression language.
- Run and debug a JSP-based application.

Accessing Resources with JNDI

- Describe the Java Naming and Directory Interface (JNDI)
- Locate or look up resources and EJBs by using:
  - JNDI APIs
  - Dependency injection

Developing the Business Logic with Session Beans

- Describe session beans.
- Create stateless and stateful session beans by using annotations.
- Describe the passivation and activation of stateful session beans.
- Use interceptor methods and classes.

Developing the Persistence Layer with JPA Entities

- Create a JPA entity.
- Select a primary key field.
- Perform O-R mapping by using annotations.
- Map inheritance between entities.
- Map relationships between entities.

Manipulating JPA Entities with the `EntityManager` API

- Declare an `EntityManager` reference.
- Look up an `EntityManager` reference by using dependency injection.
- Use the `EntityManager` API to:
  - Find an entity by its primary key
  - Insert a new entity
  - Modify an existing entity
  - Delete an entity
- Execute dynamic queries with the Query API.
- Write simple JPQL queries.

Developing the Business Logic with Web Services

- Describe Web service technologies.
- Identify the role of SOAP, WSDL, and UDDI in Web services.
- Decide on the Web service development approach.
- Develop Web services by using the top-down approach with JDeveloper.

Developing the Web Interface Using JavaServer Faces

- Describe the purpose of JSF.
- Use JSF components.
- Explain the use of managed beans.
- Describe the life cycle of a JSF application.
- Explain the role of the JSF tag libraries.
- Describe how JDeveloper supports JSF.
- Create a JSF-based JSP in JDeveloper.

Planning Navigation and Page Flow

- Use a JSF Navigation diagram to plan the pages of an application and the navigation between them.
- Describe JSF Navigation.
- Describe the types of JSF Navigation cases.
- Create a backing bean for a JSF page.

Handling Application Events

- Define the types of JSF events.
- Create event listeners for a JSF application.
- Describe how the JSF life cycle handles validation.
- List the types of validation provided by JSF.

Asynchronous Communication with

- Identify the features of a messaging system.
- Describe the Java Message Service (JMS) architecture.
- Configure a Java Message Service.
- Create a Message-Driven Bean (MDB).
- Create a JMS/MDB client.

Managing Transactions with Session and Message-Driven Beans

- Choose the appropriate type of transaction management.
- Set the transaction attribute for container-managed transactions.
- Create transaction demarcations.

Securing Java EE Applications with JAAS

- Explain the Java EE application security design.
- Describe the Java Authentication and Authorization Service (JAAS).
- Implement JAAS security for Web applications and Enterprise JavaBeans (EJB).

Packaging and Deploying Java EE Applications

- Deploy Java EE applications to the WebLogic server environment.
- Deploy applications by using:
  - Console
  - Command-line
  - JDeveloper

Troubleshooting Java EE Applications

- Use JDeveloper and tools for logging and diagnostics.
- Test a business service in isolation from views or controllers.
- Make use of FileMon, JUnit, and HTTP Analyzer.

Summary

In this lesson, you should have learned to:

- Identify the course objectives
- Discuss the course agenda
- Discuss the contents covered in the course