

This intense four-day course teaches Java programmers how to develop enterprise applications using the ease of development features introduced in Java EE 5. Students will learn how to create dynamic web applications with JSP, Java Servlets, JSTL, and JSF. They will use JAX-WS to develop SOAP based web services. Students will learn about session and message-driven EJBs, as well as the new Persistence API. They will also be introduced to JavaMail, Java Message Service, Java Transaction API, and Java Management Extensions.

Course Objectives:

- Describe the architecture and use of JEE technologies.
- Write web applications that combine Java Servlets, JavaServer Pages, and JavaBeans using the Model-View-Controller architecture.
- Develop web applications using the component-based and event-driven JavaServer Faces framework.
- Send and receive asynchronous messages with the Java Message Service.
- Interface with mail servers with JavaMail.
- Describe the Enterprise JavaBean architecture.
- Write session and message-driven EJBs.
- Access a relational database with the Java Persistence API.
- Handle transactions with the Java Transaction API.
- Integrate legacy systems with JAX-WS web services.

Audience: Java programmers who need to learn about Java EE 5.

Prerequisites: *Java Programming*

Number of Days: 4 days

<p>1. Course Introduction Course Objectives Overview Suggested References</p> <p>2. Introduction to Java EE 5 Introduction to Java EE Java SE Building Blocks Servlets, JSPs, and Web Applications Web Services Enterprise JavaBeans Additional Java EE APIs Java EE Clients POJO, Dependency Injection, and Annotations The Java EE Platform</p> <p>3. Introduction to JSP and JSTL MVC and Web Applications</p>	<p>JSP As the View JSP Scripting Elements Expression Language EL Operators Request and Response include and forwards JSTL Conditionals and Iteration in JSTL JSTL Variables And Output</p> <p>4. Introduction to Java Servlets and JavaBeans Java Servlets as the Controller HttpServlet HttpServletRequest HttpServletResponse HttpSession</p>
---	---

- RequestDispatcher
- JavaBeans as the Model
- Bean Scopes
- web.xml
- 5. JavaServer Faces**
 - Frameworks
 - JSF Benefits
 - JSF Tag Libraries
 - Components
 - Managed Beans
 - Event handling
 - Navigation
 - Validators and Converters
 - Lifecycle
 - JSF Application Structure
- 6. JMS**
 - Messaging Concepts
 - What is JMS ?
 - Point-to-Point
 - Publish/Subscribe
 - Message Object
 - Session
 - Creating the Client
- 7. JavaMail**
 - Mail Systems and JavaMail
 - The javax.mail Packages
 - Establishing a Session
 - The MimeMessage Class
 - Sending a Message
 - Retrieving Email Messages
 - Multi-part Messages
- 8. EJB3 Overview**
 - The Enterprise JavaBean
 - EJB Benefits
 - Defining the Bean Interface
 - Defining the Bean Class
 - Creating a Client Servlet
 - Assembly and Deployment of EJBs
- 9. Session Beans and Message-Driven Beans**
 - A Session Bean
 - Stateless Session Beans
 - PostConstruct and PreDestroy
 - Lifecycle of a Stateless Session Bean
 - Stateful Session Beans
 - Lifecycle of a Stateful Session Bean
- Lifecycle Callbacks
- Dependency Injection
- Message-Driven Beans
- MDB Lifecycle
- Sending a Message
- 10. Introduction to the Persistence API**
 - What is Java Persistence?
 - Persistence Objects and Metadata
 - Creating an Entity Class
 - The Entity Manager
 - Looking up Entities
 - The Persistence Unit
 - Deployment
- 11. Persisting Entities**
 - EntityManager and Persistence Context
 - Entity Lifecycle
 - Creating and Removing Entities
 - Transactions
 - Mapping Entities to Tables
 - Entity Relationships
 - Primary Keys
 - Lazy Loading and Cascading
- 12. Transactions**
 - Transaction Terminology
 - The Java Transaction API
 - The UserTransaction Interface
 - Transactions in Java EE
 - Bean-Managed Transactions
 - Container-Managed Transactions
 - Transaction Attributes
 - Transaction Rollbacks
- 13. Data Binding with JAXB 2.0**
 - W3C XML Schema
 - XML Data Binding Basics
 - JAXB Architecture
 - Compiling Schema to Java
 - JAXBContext
 - Unmarshalling
 - Marshalling
 - Validation
 - Custom Binding Declarations
 - Java to Schema

- 14. **Java API for XML-Based Web Services (JAX-WS)**
 - JAX-WS
 - Creating a Web Service Endpoint
 - The Service Implementation
 - The Service Interface
 - apt and wsgen
 - Generated Files
 - Packaging and Deploying the Application
 - A JAX-WS Client
 - wsimport
- 15. **Java Management Extension (JMX)**
 - What is a JMX?
 - MBeans
 - Creating a Standard MBean
 - Object Names
 - The MBean Server
 - Local Client
 - Remote Client
 - JConsole
 - Notifications
- 16. **Case Study**
 - Persistence
 - Stateless Session Bean
 - Web Tier Client: HTML
 - Web Tier Client: Controller Servlet
 - Web Tier Client: Data Transfer
 - JavaBean
 - Web Tier Client: JSP
 - Web Tier Client: web.xml
 - Message-Driven Bean
 - JMS Client
 - JAX-WS Endpoint
 - Web Service Client
- 17. **Appendix A – Underlying Technologies: RMI, JNDI, and JDBC**
 - RMI
 - Steps to Create a Remote Object
 - An RMI Client
 - An RMI Server
 - RMI Utilities
 - JNDI Naming and Directory Services
 - Namespaces and Contexts
 - Naming Operations

- Bindings
- JNDI in JAVA EE
- The JDBC Connectivity Model
- Connecting to the Database
- Creating a SQL Query
- Getting the Results
- Updating Database Data
- 18. **Appendix B – Ant**
 - What Is Ant?
 - build.xml
 - Tasks
 - Properties and Property Files
 - Managing Files and Directories
 - Filesets
 - Java Tasks
 - Creating Java Archives
 - Specifying Paths
 - Miscellaneous Tasks