

# Enterprise Java Course Outline

**Course Length:**

5 days

**Audience:**

This course is intended for programmers who have a good working knowledge of the Java programming language, as well as a basic understanding of Web technologies, HTML, distributed computing, and JDBC.

**Objectives:**

- ❑ Become familiar with the enterprise Java technologies of J2EE.
- ❑ Understand what CORBA provides.
- ❑ Develop a CORBA application.
- ❑ Understand the architecture of Java servlets.
- ❑ Develop a servlet application.
- ❑ Learn how to deploy a servlet to a Web server.
- ❑ Connect to a database from within a servlet.
- ❑ Understand the goals of JavaServer Pages.
- ❑ Learn how to write scriptlets using JavaServer Pages.
- ❑ Learn how to use JavaBeans from within a JavaServer Page.
- ❑ Understand the goals of XML.
- ❑ Understand how XML compares to HTML and SGML.
- ❑ Write an XML document.
- ❑ Use the Simple API for XML (SAX).
- ❑ Use the Document Object Model (DOM).

## Overview of J2EE Technologies

- ❑ **The J2EE Specification:** A superset of the EJB specification.
- ❑ **The EJB Specification:** A description of services provided by an EJB server.
- ❑ **Java Servlets:** Applications that run in a Web server.
- ❑ **Java Database Connectivity:** Connecting to a database from within a Java application.
- ❑ **Java Server Pages:** Used for creating dynamic Web page content.
- ❑ **JNDI:** The Java Naming and Directory Interface.
- ❑ **JTS:** Java Transaction Service and the Java Transaction API.
- ❑ **Java Message Service:** An API for enterprise messaging needs.
- ❑ **Java IDL:** The Interface Definition Language and implementing CORBA with Java.
- ❑ **XML:** The extensible markup language.
- ❑ **Java Mail:** An overview of the Java Mail API.

## Servlets

- ❑ **Overview of Servlets:** Programs that run in a Web server.
- ❑ **Server-side Development:** Options for creating Web applications.
- ❑ **Advantages of Servlets:** Powerful, portable and secure.
- ❑ **The Servlet API:** The javax.servlet and javax.servlet.http packages.
- ❑ **HTTP:** The Hypertext Transfer Protocol.
- ❑ **Requests:** The HTTP Request header.
- ❑ **Responses:** The HTTP Response header.
- ❑ **HTML Basics:** A quick introduction to HTML and its syntax.
- ❑ **HTTP Servlets:** The HttpServlet class.
- ❑ **The “Hello, World” Servlet:** A simple HTTP servlet handling a GET request.
- ❑ **The Lifecycle of a Servlet:** The servlet container.
- ❑ **A Servlet Counter:** A demonstration of the lifecycle of a servlet.
- ❑ **HttpServletRequest:** Encapsulating the HTTP request header of a client request.
- ❑ **HttpServletResponse:** Encapsulating the HTTP response to a client.
- ❑ **Parameters:** Obtaining data from the client request.
- ❑ **The Request Dispatcher:** Including servlets and forwarding requests to other servlets.
- ❑ **Scope:** Understanding the various scopes an object can have within a servlet container.
- ❑ **Request Scope:** Objects associated with a client request.
- ❑ **Application Scope:** A container-wide object.
- ❑ **Session Scope:** Creating client sessions.
- ❑ **Cookies:** A sweet session tracker.
- ❑ **The HttpSession Interface:** Container session objects.
- ❑ **URL Rewriting:** Encoding a URL.
- ❑ **Servlet Threading Models:** The SingleThreadModel.
- ❑ **Connecting to a Database:** JDBC within a servlet.
- ❑ **Server-side Includes:** A servlet embedded in an HTML page.

## JavaServer Pages

- ❑ **Server-side Development:** Comparing JSP to other technologies.
- ❑ **Overview of JavaServer Pages:** Java technology for creating dynamic Web content.
- ❑ **JSP Lifecycle:** Translation of JSP into servlets.
- ❑ **An Introduction to Servlets:** Understanding how servlets run on a Web server.
- ❑ **“Hello, JSPs”:** A simple JavaServer Page.
- ❑ **JavaServer Page Tags:** The various JSP tags.
- ❑ **Templating:** The include directive.
- ❑ **Declarations:** Declaring variables and methods.
- ❑ **Expressions:** Java statements within a JSP.
- ❑ **The request Object:** The request header information.
- ❑ **Parameters:** Request data passed in to the JSP.
- ❑ **Scriptlets:** Embedding Java code within a JSP.
- ❑ **Directives:** Specifying options for the page.
- ❑ **Implicit Objects:** The implicit objects of a JSP.
- ❑ **JSP Actions:** The action tags of JSP.
- ❑ **Overview of JavaBeans:** Java software components.
- ❑ **JavaBeans in JSPs:** The useBean directive.
- ❑ **Error Pages:** Handling exceptions in JSP.
- ❑ **Scope:** The four scopes of JSP objects.
- ❑ **Sessions:** Creating sessions within JSP.
- ❑ **Request Scope:** Associating data with the request.
- ❑ **Application Scope:** The implicit application object.
- ❑ **Session Scope:** Creating client sessions.
- ❑ **The Implicit Session Object:** Session tracking made easy.
- ❑ **Beans with Session Scope**

## Java and CORBA

- ❑ **The Object Management Group:** Understanding the goals of the OMG.
- ❑ **Overview of CORBA:** A specification for distributed objects.
- ❑ **The Java IDL:** The Interface Definition Language.
- ❑ **The ORB Architecture:** Handling requests from a client.
- ❑ **Services of the ORB:** Naming, transaction, and event services.
- ❑ **The Portable Object Adapter:** Communicating with the servant.
- ❑ **The Interface Definition Language:** The IDL keywords.
- ❑ **Writing an IDL Interface:** The interface and attribute keywords.
- ❑ **Adding Methods:** Declaring the methods of an interface.
- ❑ **Compiling the IDL:** Generating the stubs and helper classes.
- ❑ **Defining Exceptions:** The exception keyword.

- ❑ **Implementing CORBA on the Server:** The steps involved in implementing CORBA on the server.
- ❑ **Step 1:** Implement the IDL interfaces.
- ❑ **Step 2:** Initialize the ORB.
- ❑ **Step 3:** Create the POA.
- ❑ **Step 4:** Generate an IOR for the servant.
- ❑ **Step 5:** Run the ORB.
- ❑ **The Client Application:** Implementing CORBA on the client side.
- ❑ **The Naming Service:** Locating objects using a naming hierarchy.
- ❑ **COS Naming:** The CORBA object naming service.
- ❑ **Using the Naming Service:** Binding and resolving objects.

## XML

- ❑ **Overview of XML:** Understanding the goals of XML.
- ❑ **XML Documents:** Well-formed vs. valid.
- ❑ **Using XML:** The components of an XML document.
- ❑ **The Root Element:** The highest-level of the document.
- ❑ **Elements:** Describing your data with element tags.
- ❑ **Creating a DTD:** Adding constraints to XML.
- ❑ **XML APIs:** SAX and JAXP.
- ❑ **Selecting a Parser:** Loading the parser class.
- ❑ **Parsing an XML Document:** The XMLReader interface.
- ❑ **Content Handlers:** Handling callbacks during parsing.
- ❑ **Error Handlers:** Handling errors and warnings during parsing.
- ❑ **DOM:** The Document Object Model.
- ❑ **Creating DOM Objects:** The Document interface.
- ❑ **Using DOM:** Creating an XML document using DOM.