J2EE PROGRAMMING WITH WEBSPHERE STUDIO APPLICATION DEVELOPER (WSAD) V5.1.1

Course Duration: 5 Days

Course Overview: This course teaches programmers how to write enterprise applications based on the Java 2 Enterprise Edition (J2EE) architecture. In addition to the core concepts of J2EE, this course provides specific instruction on building J2EE applications using WebSphere Studio Application Developer v5.1.1 (WSAD), targeting deployment in WebSphere Application Server v5 (WAS).

Topics Covered:

- Core J2EE technology concepts: Servlets, JSP, EJB
- Servlet and JSP programming with WebSphere Studio Application Developer
- EJB programming with WebSphere Studio Application Developer
- Debugging J2EE applications

LEARNING OBJECTIVES

- After completing this course, the student should be able to:
- The core concepts of J2EE: Servlets, Java Server Pages, Enterprise Java Beans
- How to develop and maintain J2EE applications
- How to leverage the power of WSAD v5.1.1 to build and test J2EE applications quickly and effectively
- How to deploy J2EE applications into WebSphere Application Server

Who Should Attend: This course is designed for Java programmers and web application designers who have interest in learning how to build or maintain enterprise applications based on the J2EE architecture using IBM's WebSphere tools.

Prerequisites: Students are expected to have a good understanding of web technology and server-side programming for dynamic web content. Experience with Java is recommended, but even a beginner's knowledge of Java is sufficient.

COURSE OUTLINE

Introduction
- WebSphere Studio Site Developer
- WebSphere Studio Application Developer
- Major differences with Visual Age for Java

Introduction to J2EE and WebSphere v5.1
- J2EE-based Web application architecture
- Application server functionality
- WebSphere platform introduction

Overview
- Perspectives and views
- WSAD J2EE development
- WSAD project concepts
- EJB project
- Web project
- The WebSphere Test Environment in WSAD

Servlet Basics
- Servlet overview
- HTTP Servlet API
- Servlet life cycle
- Example
- Developing servlets

JSP Basics
- JSP overview
- JSP directives
- JSP expressions
- JSP scriptlets
- JSP declarations
- Examples
- Developing and testing JSP

Servlet Interaction and Advanced Servlets
- Request
- Response
- Servlet Context
- Session

Servlet Programming - Advanced
- Programming with the MVC model
- JDBC and database connection pooling
- Session tracking
- Integrating servlet and JSP
- Examples

JSP Techniques
- Predefined variables
- Using JavaBeans with JSP
- Examples

Using Java Beans with JSP
- Present the concept of a Java Bean
- Describe using Java Beans in a JSP
- Mixing Scriplets and Bean tags

Enterprise JavaBeans Programming - Overview
- EJB fundamentals
- Overview of EJB container and EJB server responsibilities
- Overview of entity EJBs and session EJBs
- EJB deployment descriptor

Enterprise JavaBeans Programming - Session Bean
- Session bean contexts and lifecycle
- Writing stateless session beans
- Writing stateful beans
- Examples
- Creating session beans

Enterprise JavaBeans Programming - Entity Bean
- Entity bean contexts and lifecycle
- Writing container-managed entity beans
- Writing bean-managed entity beans
- Examples
- Creating entity beans

Best Practices

- Stateful Session Beans
- Transaction Isolation Levels
- Which EJB objects to cache
- Deploying the Web and EJB containers
- CMP EJB Pooling
- Access Beans
- Using HTTP Session
- Connection Pooling
- Java Best Practices
- WebSphere Best Practices

Deploying Enterprise Application Projects

- Exporting the Enterprise Application
- Deployment Descriptors
- Application Assemble Entries
- Installing an Application into WAS v5
- Managing WAS applications

J2EE Design Patterns

- EJB Layer Architectural Patterns
- Inter-tier Data Transfer Patterns
- Transaction and Persistence Patterns
- Client-side EJB Interaction Patterns
- Primary Key Generation Strategies

Using WSAD in a Large Project

- Performance tuning
- Breaking up large projects
- Remote debugging

Appendix A: Message-Driven Beans

Appendix B: Custom JSP Tag Libraries

Appendix C: Java Messaging Service