

**Course Description:** This course is a comprehensive introduction to writing Enterprise JavaBeans (EJB) using IBM WebSphere and the Rational Application Developer (RAD) tool. An overview of J2EE technology is provided, followed by hands-on experience with JNDI, JDBC, JMS, session beans, entity beans, and message-driven beans. The EJB 2.x specification is covered, with emphasis on container-managed persistence (CMP) and container-managed relationships (CMR).

**Who Should Attend:** This course is for experienced Java programmers and software engineers preparing to write Enterprise JavaBeans for J2EE applications hosted on IBM WebSphere Application Server.

**Prerequisites:** Students should be comfortable with Java programming and object-oriented concepts. A minimum of six months coding experience is suggested. In addition, students should have prior experience using JDBC and SQL.

**Benefits of Attendance:** Upon completion of this course, students will be able to:

- Start, stop, and configure IBM WebSphere Application Server
- Use JNDI to access database and EJB resources
- Create JDBC data sources
- Write stateless and stateful session beans
- Use bean-managed and container-managed persistence
- Understand and write XML-based deployment descriptors
- Configure and deploy EJBs using the Rational Application Developer tool
- Assemble J2EE-compliant enterprise applications
- Use the Session Facade pattern
- Implement container-managed relationships
- Create JMS destinations
- Write message-driven beans

## Course Outline:

### Overview of Java EE

Java Platforms  
Characteristics of "Enterprise" Computing  
Java EE Technologies  
Multi-Tier Architectures  
Advantages of Multi-Tier Architectures  
Container-Based Approach  
Parties Involved in Java EE Deployment  
Java EE-Compliant Application Servers  
Java EE Application Models  
HTTP Services Application Model  
N-Tiered Application Model

### Introduction to RAD

What is WebSphere@?  
What is Rational@ Application Developer?  
Starting RAD  
Creating a Java Project  
Importing Existing Java Code  
Creating a New Java Program

### Java Naming and Directory Interface

What is JNDI?  
Benefits of JNDI  
Naming Services  
Directory Services  
Using JNDI  
Context Operations  
JNDI Utility Class  
JNDI Example  
Naming Exceptions  
Creating a Server Instance  
Starting and Stopping WebSphere  
Running the JNDI Example

### Using JDBC Data Sources

A Simple JDBC Program  
JDBC Driver Types  
Using the Derby Database  
JDBC Data Sources  
Data Source Example  
Configuring a JDBC Provider  
Configuring a Data Source  
Running the JDBC Examples  
Executing a Query  
Using the Database Explorer

### RMI and IOP

Object Serialization  
Remote Method Invocation

RMI Architecture  
The Remote Interface  
CORBA

### Enterprise JavaBeans

Enterprise JavaBeans Component Model  
Types of Enterprise Beans  
EJB Wrapper Interfaces  
Deployment Descriptors  
Context and Environment Objects  
EJB Runtime Environment  
The Remote Interface  
The Home Interface  
The Enterprise Bean Class  
The Client Test Program  
The ejb-jar.xml File  
The ibm-ejb-jar-bnd.xml File  
Creating an Enterprise Application Project  
Deploying the Enterprise Application  
Testing with the IBM Universal Client

### Session Beans

Session Bean Lifetime  
Session Bean Interface  
Session Bean Lifecycles  
Stateless Session Bean Example  
Accessing Environment Entries  
Stateful Session Bean Example  
EJB Exceptions - Examples  
Testing the Session Beans  
Creating a New Session Bean

### BMP Entity Beans

Entity Beans  
Entity Bean Interface  
Lifecycle of an Entity Bean  
Bean-Managed Persistence Example  
Deploying Entity Beans  
Deployment Settings for BMP Entity Beans

### CMR Entity Beans

Container-Managed Persistence  
Primary Key Class  
Implementing CMP Entity Bean Methods  
Container-Managed Persistence Example  
Deployment Settings for CMP Entity Beans  
Deployment Settings for Custom Finders  
EJB Query Language  
Mapping Container-Managed Fields  
Testing the Product Bean

### Session Facade Pattern

J2EE Design Patterns  
Session Facade Pattern  
Local Interfaces  
Example - ItemOrderer Bean  
Deployment Settings for ItemOrderer Bean  
Testing the Session Bean  
Bottom-Up Mapping  
Configuring the Application Client Project

### Container-Managed Relationships

Container-Managed Relationships  
Container-Managed Relationship (CMR) Example  
CMR Example - Local Interfaces  
CMR Example - Local Home Interfaces  
CMR Example - Entity Bean Classes  
Transfer Object Pattern  
CMR Example - Session Bean  
CMR Example - Deployment Descriptors  
Creating New CMP Entity Beans  
Creating a Relationship  
Generating a Top-Down Mapping  
Creating the Tables  
Adding an Existing Session Bean  
Running the Client Program

### Java Message Service

Introduction  
JMS and the J2EE Platform  
Basic JMS Concepts  
The JMS Programming Model  
Point-to-Point Example - Sender  
Point-to-Point Example - Receiver  
Configuring JMS for WebSphere  
Running the Point-to-Point Example  
Publish/Subscribe Example - Publisher  
Publish/Subscribe Example - Subscriber  
Running the Publish/Subscribe Example  
Reliable Message Delivery

### Message-Driven Beans

Message-Driven Beans  
Message-Driven Bean Lifecycle  
Message-Driven Bean Example  
Configuring an Activation Specification  
Deploying Message-Driven Beans  
Creating a New Message-Driven Bean

### Appendix A: Web Resources

Java Technology  
WebSphere  
Derby Database

### Appendix B: Using EJBs in a Web Application

Using Web Components as EJB Clients  
Servlet Code for the Survey Application  
Session Bean for the Survey Application  
Deploying the Survey Application

### Appendix C: EJB Transactions

Transactions  
Container-Managed Transactions  
Transaction Attributes  
System vs. Application Exceptions  
Rolling Back a Container-Managed Transaction  
Configuring a Transactional Data Source

### Appendix D: EJB Security

Java EE Security  
Specifying Permissions for EJBs  
Enabling Security for WebSphere

### Appendix E: EJB Timer Service

Overview of the Timer Service  
Timer Service API  
Creating Timers  
Canceling and Saving Timers  
Example  
Running the Example

### Appendix F: Introduction to EJB 3

Limitations of EJB 2  
EJB 3 Feature Overview  
Comparing EJB 2 and 3  
The EJB 3 Business Interface  
The Annotated EJB Class  
Dependency Injection  
Container Callback Methods  
Stateless Session Beans  
Stateful Session Beans  
Entity Beans