# Advanced Business Component Development With Enterprise JavaBeans™ Technology (SL-351-EE5)

## Overview

The Business Component Development with Enterprise JavaBeans Technology course provides students with the knowledge of how to develop robust back-end functionality using Enterprise JavaBeans (EJB) version 3.0 technology. This course uses an online auction scenario to demonstrate how to leverage container-managed services with session and message beans, and entity classes to resolve the real-world problems presented by an electronic commerce application.The emphasis of this course is on providing practical EJB technology coding experience, while also covering the designs and best practices used to solve transaction, messaging, and security issues. The course features the Java Platform, Enterprise Edition 5 (Java EE 5) technology, and uses the Java EE 5 SDK. The students perform the course lab exercises using the NetBeans Integrated Development Environment (IDE) 5.5. The hands-on lab environment uses the Java EE reference implementation server to provide students with a non-vendor-specific experience.

## Who Can Benefit

EJB technology business component developers who are tasked with the following responsibilities:

- Analyzing, designing, developing, and testing an EJB technology applications
- Deploying an EJB technology application
- Integrating an EJB technology application with legacy application

## Required Prerequisites:

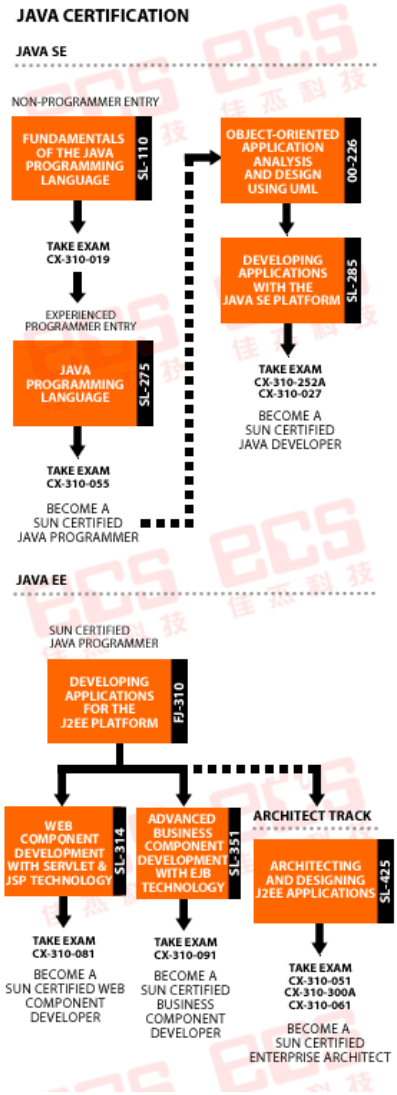
- Display experience with the Java programming language
- Integrate existing Java code (for example, reuse existing classes created by other team members)
- Design Java technology applications
- Although not required, experience with distributed computing concepts in Java technology is an advantage
- Java Programming Language, Java SE 6 (SL-275-SE6)

## Skills Gained

After successful completion of this course, the student will be able to:

- Implement business-tier functionality using EJB technology
- Describe best practices and other advanced issues in business component development with EJB technology
- Assemble and deploy EJB technology business-tier components on an application server
- Integrate an EJB technology-based application using the Java Messaging Service API
- Create and use Query objects using the Java Persistence Query Language

## Roadmap




## Certification Alignment

- Sun Certified Business Component Developer, SCBCD

## Suggested Next Courses:

- Developing Architectures for Enterprise Java Applications (SL-425)
- Java EE 5 Patterns (SL-500-EE5)

# Advanced Business Component Development With Enterprise JavaBeans™ Technology (SL-351-EE5)

## Content

### Examining EJB Applications
- Introduce the Java Platform, Enterprise Edition (Java EE)
- Examine the Java EE application architecture
- Examine the Java EE application creation process

### Introducing the Auction Application
- Describe the auction application
- Define the domain objects of the auction application
- Describe the implementation model for the auction system

### Implementing EJB 3.0 Session Beans
- Create session beans: Essential tasks
- Create session beans: Add life-cycle event handlers
- Package and deploy session beans
- Create a session bean client

### Implementing Entity Classes: The Basics
- Define entity classes: Essential tasks
- Manage the life-cycle of an entity instance
- Define entity beans: Add life-cycle event handlers
- Package and deploy entity classes

### Implementing Entity Classes: Modelling Data Association Relationships
- Examine association relationships in the data and object models
- Use relationship properties to define associations
- Implement unidirectional and bidirectional associations

### Implementing Entity Classes: Modelling Inheritance Relationships
- Examine entity class inheritance
- Inherit from an entity class
- Inherit from an abstract entity class
- Inherit from a non-entity class
- Inherit using an embedded superclass
- Examine Inheritance mapping strategies
- Define entity classes: Using an embedded class
- Define entity classes: Using a composite primary key

### Using the Java Persistence Query Language (QL)
- Examine the Java Persistence query language
- Create and use the SELECT statement
- Create and use the BULK UPDATE statement
- Create and use the DELETE statement
- Create and use Query objects

### Developing Java EE Applications Using Messaging
- Describe the roles of the participants in the JMS API messaging system
- Write a message producer
- Write an asynchronous message listener
- Write a synchronous message listener
- List the messaging capabilities and limitations of session, entity, and message-driven beans

### Developing Message-Driven Beans
- Describe the properties and life cycle of message-driven beans
- Create a JMS message-driven bean
- Create a non-JMS message-driven bean

### Implementing Interceptor Classes and Methods
- Create a business interceptor method in the enterprise bean class
- Create an interceptor class
- Associate multiple business interceptor methods with an enterprise bean
- Include life-cycle callback interceptor methods in an interceptor class
- Create entity life-cycle callback methods

### Implementing Transactions
- Describe the transaction demarcation task
- Implement Container-Managed Transactions (CMT)
- Interact programmatically with an ongoing CMT transaction
- Implement Bean-Managed Transactions (BMT)
- Apply transactions to messaging

### Handling Exceptions
- Introduce exceptions in Java EE applications
- Describe the exception path in a Java EE application environment
- Describe EJB container exception handling
- Handle exceptions in an enterprise bean's methods
- Handle exceptions in an enterprise bean's client code
- Review specific issues relating to exception handling in EJB technology applications

### Using Timer Services
- Create a timer callback notification
- Process a timer callback notification
- Manage timer objects

### Implementing Security
- Understand the Java EE security architecture
- Authenticate the caller
- Examine Java EE authorization strategies
- Use declarative authorization
- Use programmatic authorization
- Examine the responsibilities of the deployer

### Using EJB Technology Best Practices
- Define best practices and state the benefits of using EJB technology best practices
- Select and apply known patterns to Java EE application design