

Oracle Java Trainings & Certifications

Binnur Kurt, Ph.D.

Omega Training and Consultancy

www.omegaegitim.com

binnur.kurt@gmail.com

Background

1995, B.Sc., ITU Computer Engineering

1997, M.Sc., ITU Computer Engineering

2007, Ph.D., ITU Computer Engineering

1995-2003, Research Assistant, *ITU CE*

2004-2008, Lecturer, ITU CE

2008-to date, Trainer&Consultant,

Omega Training and Consultancy

2008-to date, Visiting Faculty Member, *ITU CE*

2004-to date, *Ahmet Yesevi Üniversitesi, Uzaktan Eğitim*

Areas of Interest

- > Java Desktop and Enterprise Technologies
- > SOA
- > OOP, OOAD
- > MySQL
- > Oracle Directory Server
- > Oracle Middleware Technologies
- > Digital Image & Video Analysis and Processing
- > Real-Time Computer Vision Systems

Agenda

1. Java Platforms (JDK 7, EE 6)
2. Oracle Java Trainings
3. Oracle Java Certifications

1

Java Platforms

What is Java Technology

- > A programming language
- > A development environment
- > An application environment
- > A deployment environment

Java Programming Language

- > General-purpose, concurrent, strongly typed, class-based object-oriented language.



James Gosling

- > Defined by the Java language specification.
- > The primary building block of a Java technology application is a class.

Object Oriented Programming (OOP)

- > Problem: Developing High Quality Software under certain constraints such as
 - Time
 - Budget
 - HR,...
- > OOP helps in developing HQS through
 - > Encapsulation – Information Hiding
 - > Inheritance - Reusability
 - > Polymorphism - Agility

Software Lifecycle

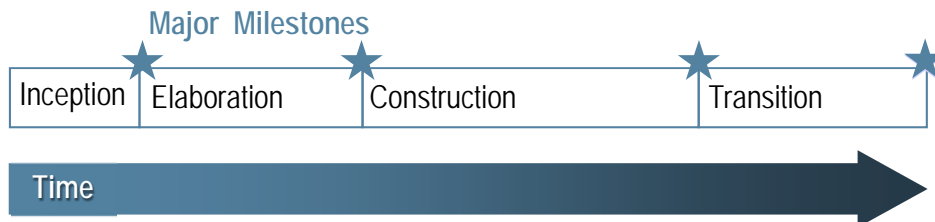
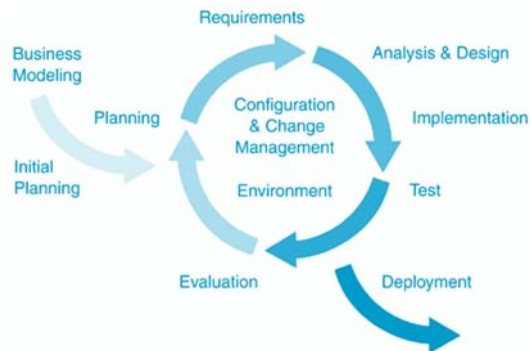
- > Requirement Analysis
- > Software Analysis
- > Design
 - Architectural Design
 - Detailed Design
- > Coding
- > Unit Testing
- > Integration Testing
- > Deployment
- > Maintenance

Software Lifecycle Management

- > No management?
- > Waterfall
- > Spiral Model
- > UP
- > XP
- > ...

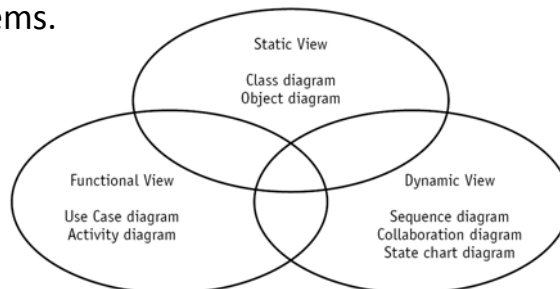
UP

- > Iterative
- > Incremental
- > Architecture Centric
- > Risk focused
- > Use case driven



UML (Unified Modeling Language)

- > A standard language for
 - specifying,
 - visualizing,
 - constructing,
 - and documenting
- > the artifacts of software systems, as well as for business modeling and other non-software systems.

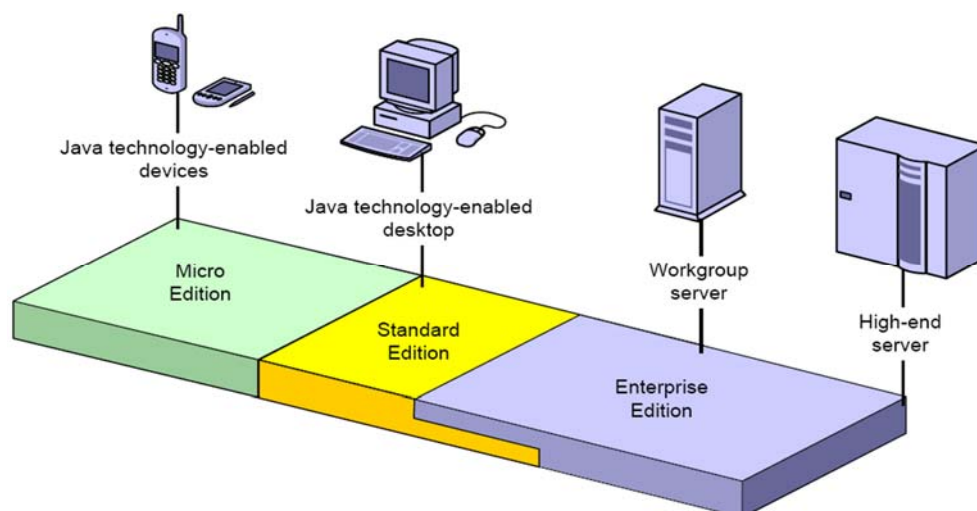


OOP does not guarantee HQS

> You need other paradigms, frameworks, tools, ...

- GP
- AOP
- DI
- Spring

Java Platforms



Java SE (Standard Edition)

- > Java Platform, **Standard Edition (Java SE)** lets you develop and deploy Java applications on **desktops** and **servers**

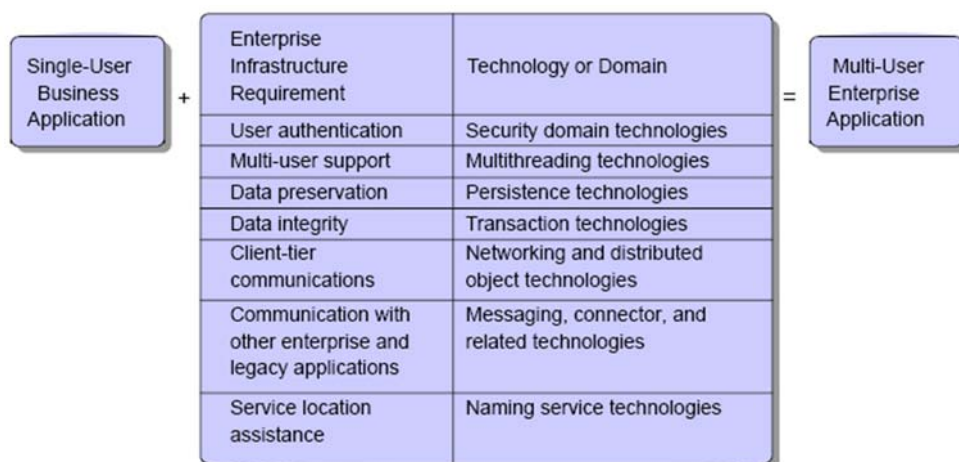
Java ME (Micro Edition)

- > Java Platform, **Micro Edition (Java ME)** provides a robust, flexible environment for applications running on mobile and other embedded devices
 - Mobile phones,
 - Personal digital assistants (PDAs),
 - TV set-top boxes,
 - Printers,...

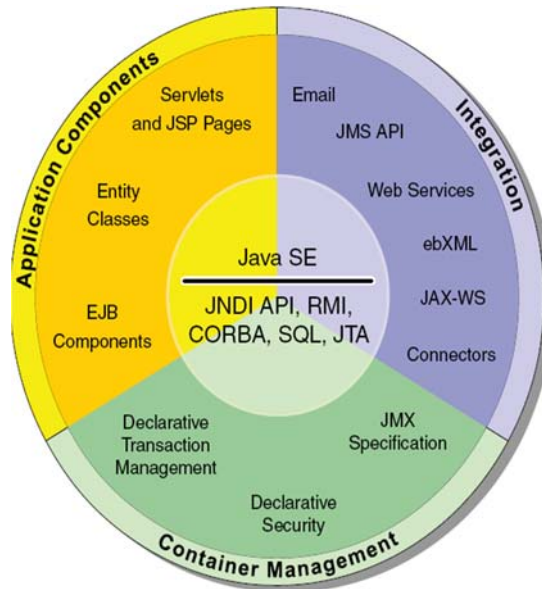
Java EE (Enterprise Edition)

- > Is an architecture for implementing enterprise-class applications
- > Uses Java and Internet technology
- > Has a primary goal of simplifying the development of enterprise-class applications through an application model that is:
 - Vendor-neutral
 - Component-based

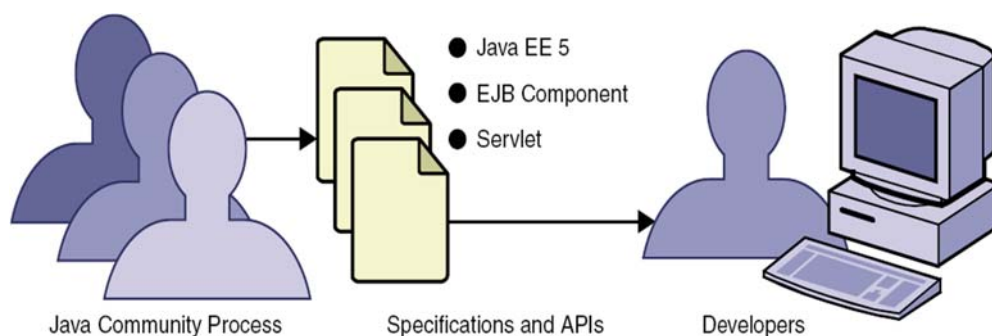
Enterprise Application Infrastructure Technologies



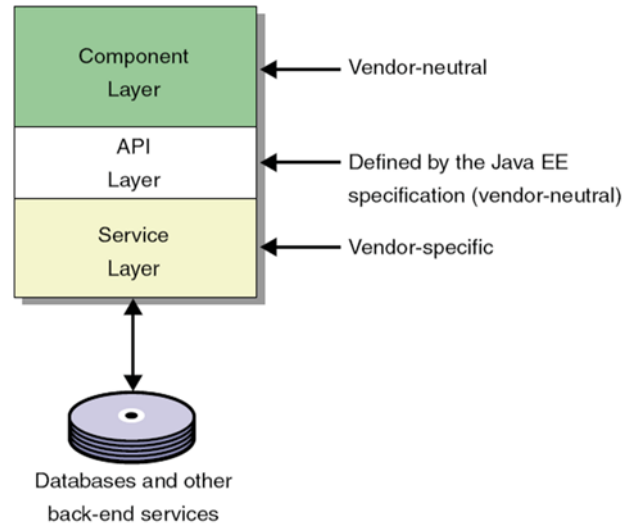
Java EE Technology Suite



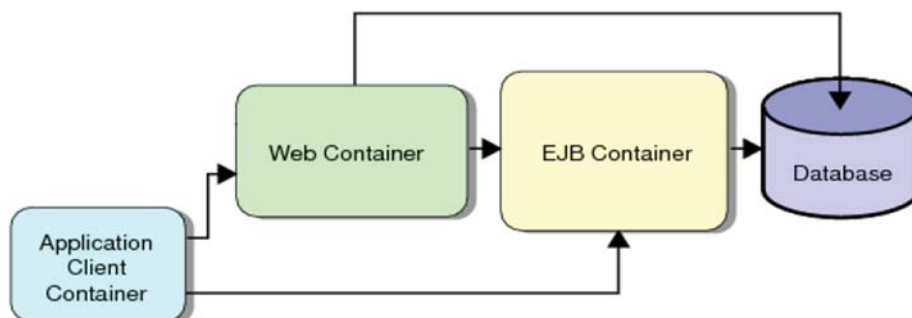
Java EE Specifications and Java Community Process (JCP)



Component, API, and Service Layer



Java EE Component Containers



Advantages of Using Server-Provided Services

Build From the Ground Up



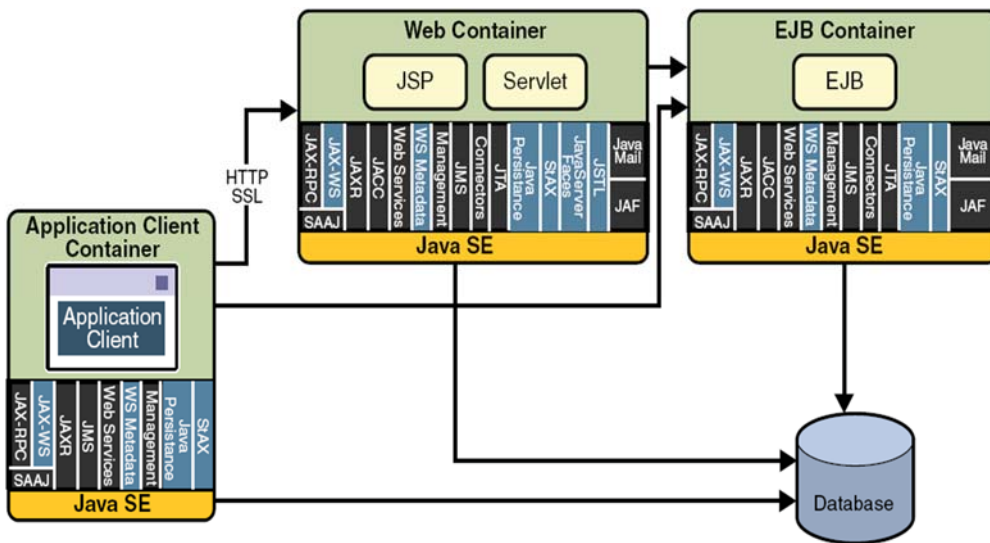
- Developer's Checklist
- Business services
 - Persistence
 - Transaction management
 - Multi-threading
 - Security management
 - Networking
 - Service publishing

Use Application Component Server



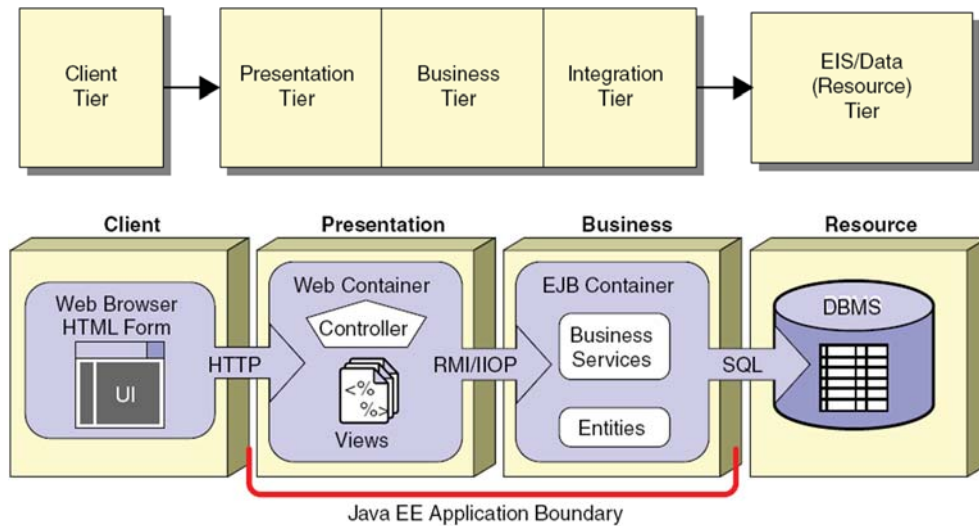
- Developer's Checklist
- Business services
- Services Provided by Server
- Persistence
 - Transaction management
 - Multi-threading
 - Security management
 - Networking
 - Service publishing

Java EE Service Infrastructure

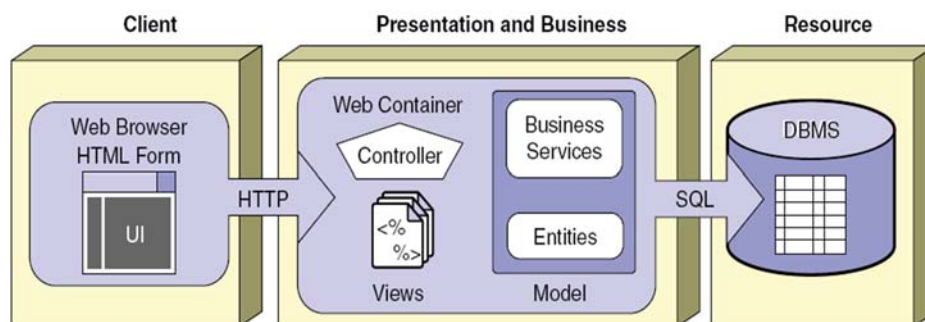


New in Java EE 5

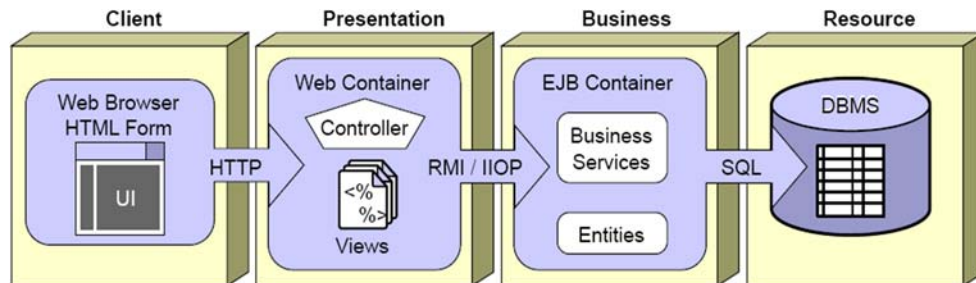
Java EE Tiered Architecture



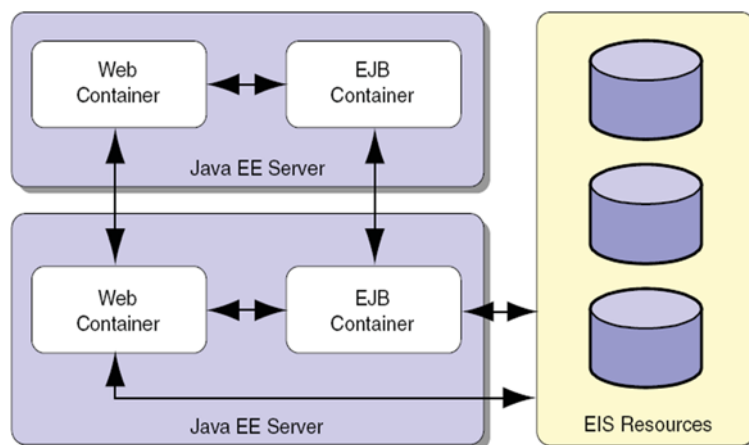
Java EE Web-Centric Architecture



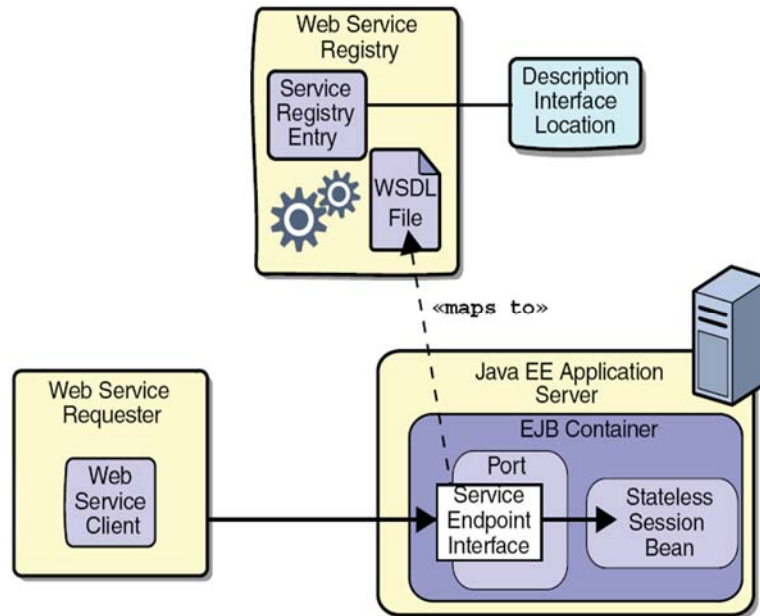
Java EE EJB Component-Centric Architecture



B2B Application Architecture



Java EE Web Service Architecture



2

Oracle Java Trainings

Sun Java Courses – Java Platform SE

SL-110-SE6	Fundamentals of the Java PL
SL-275-SE6	Java Programming Language
SL-275-SE7	Java Programming Language
SL-285-SE6	Developing Applications with Java SE Platform
D69518GC10	Java Performance Tuning and Optimization
OO-226	Object-Oriented Analysis & Design using UML

Sun Java Courses – Java Platform ME

- **DTJ-365** Developing Mobile Phone Applications With J2ME Technology

Sun Java Courses – Java Platform EE

- > **FJ-310-EE6** Developing Applications for the Java EE Platform
- > **SL-314-EE6** Web Component Development with Servlet & JSP Technologies
- > **SL-351-EE6** Business Component Development *with* Enterprise JavaBeans Technology
- > **DWS-3111-EE5** Developing Java Web Services *using* Java Technology
- > **DWS-4112-EE5** Designing Java Web Services
- > **D65185GC10** Developing Web Services Using Java Technology, Java EE6
- > **D66105GC10** Developing Secure Java Web Services, Java EE6
- > **SL-370-EE6** Building Database Driven Applications with JPA 2.0
- > **SL-340-EE6** Developing Web Application Using the JSF 2.0
- > **SL-425** Developing Architectures for Enterprise Java Applications
- > **SL-500-EE** J2EE Patterns

Java Trainings Course Contents

SL-110-SE6**Fundamentals of the Java Programming Language**

- > After: SL-275-SE6
- > Certification: Sun Certified Java Associate (SCJA)
- > **Module 1** Explaining Java Technology
- > **Module 2** Analyzing a Problem and Designing a Solution
- > **Module 3** Developing and Testing a Java Technology Program
- > **Module 4** Declaring, Initializing, and Using Variables
- > **Module 5** Creating and Using Objects
- > **Module 6** Using Operators and Decision Constructs
- > **Module 7** Using Loop Constructs
- > **Module 8** Developing and Using Methods
- > **Module 9** Implementing Encapsulation and Constructors
- > **Module 10** Creating and Using Arrays
- > **Module 11** Implementing Inheritance

SL-275-SE6**Java Programming Language**

- > After: SL-285-SE6, FJ-310-EE5
- > Certification: Sun Certified Java Programmer (SCJP)
- > **Module 1** Getting Started
- > **Module 2** Object-Oriented Programming
- > **Module 3** Identifiers, Keywords, and Types
- > **Module 4** Expressions and Flow Control
- > **Module 5** Arrays
- > **Module 6** Class Design
- > **Module 7** Advanced Class Features
- > **Module 8** Exceptions and Assertions
- > **Module 9** Collections and Generics Framework
- > **Module 10** I/O Fundamentals
- > **Module 11** Console I/O and File I/O

SL-275-SE6**Java Programming Language**

- > [Module 12](#) Building Java GUIs Using the Swing API
- > [Module 13](#) Handling GUI-Generated Events
- > [Module 14](#) GUI-Based Applications
- > [Module 15](#) Threads
- > [Module 16](#) Networking

OO-226**Object-Oriented Analysis and Design Using UML**

- > Before: SL-275-SE6 After: SL-285-SE6, SL-425, SL-500
- > [Module 1](#) Introducing the Software Development Process
- > [Module 2](#) Examining Object-Oriented Technology
- > [Module 3](#) Choosing an Object-Oriented Methodology
- > [Module 4](#) Determining the Project Vision
- > [Module 5](#) Gathering the System Requirements
- > [Module 6](#) Creating the Initial Use Case Diagram
- > [Module 7](#) Refining the Use Case Diagram
- > [Module 8](#) Determining the Key Abstractions
- > [Module 9](#) Constructing the Problem Domain Model
- > [Module 10](#) Creating the Analysis Model Using Robustness Analysis
- > [Module 11](#) Introducing Fundamental Architecture Concepts
- > [Module 12](#) Exploring the Architecture Workflow

OO-226**Object-Oriented Analysis and Design Using UML**

- > [Module 13](#) Creating the Architectural Model for the Client and Presentation
- > [Module 14](#) Creating the Architectural Model for the Business Tier
- > [Module 15](#) Creating the Architectural Model for the Resource and Integration Tiers
- > [Module 16](#) Creating the Solution Model
- > [Module 17](#) Refining the Domain Model
- > [Module 18](#) Applying Design Patterns to the Solution Model
- > [Module 19](#) Modeling Complex Object State Using Statechart Diagrams

SL-285-SE6**Developing Applications With the Java SE Platform**

- > Before: SL-275-SE6 After: FJ-310-EE5 or FJ-310-EE6
- > Certification: Sun Certified Java Developer (SCJD)
- > [Module 1](#) Introduce the BrokerTool Application
- > [Module 2](#) Apply the Model View Controller (MVC) Design Pattern
- > [Module 3](#) Implement Unit Testing
- > [Module 4](#) Design the BrokerTool Application
- > [Module 5](#) Implement the Java Database Connectivity (JDBC) API
- > [Module 6](#) Create Graphical User Interfaces (GUI)
- > [Module 7](#) Handle GUI Events
- > [Module 8](#) Log Messages in GUI
- > [Module 9](#) Implement Multiple-Tier Design
- > [Module 10](#) Implement Advanced Multiple-Tier Design
- > [Module 11](#) Communicate With Remote Objects Using Java RMI

FJ310-EE5**Developing Applications for the Java EE Platform**

- > Before: SL-275-SE6, OO-226 After: SL-314-EE5, SL-351-EE5
- > **Module 1** Placing the Java EE Model in Context
- > **Module 2** Java EE Component Model and Development Step
- > **Module 3** Web Component Model
- > **Module 4** Developing Servlets
- > **Module 5** Developing With JavaServer Pages Technology
- > **Module 6** EJB Component Model
- > **Module 7** Implementing EJB 3.0 Session Beans
- > **Module 8** The Java Persistence API
- > **Module 9** Implementing a Transaction Policy
- > **Module 10** Developing Java EE Applications Using Messaging
- > **Module 11** Developing Message-Driven Beans
- > **Module 12** Web Service Model

FJ310-EE5**Developing Applications for the Java EE Platform**

- > **Module 13** Implementing Java EE Web Services with JAX-WS
- > **Module 14** Implementing a Security Policy

FJ310-EE6**Developing Applications for the Java EE Platform**

- > Before: SL-275-SE6, OO-226 After: SL-314-EE6, SL-351-EE6
- > **Module 1** Survey of Java EE Technologies
- > **Module 2** Enterprise Application Architecture
- > **Module 3** Web Technology Overview
- > **Module 4** Developing Servlets
- > **Module 5** Developing With JavaServer Pages Technology
- > **Module 6** JavaServer Faces
- > **Module 7** EJB Overview
- > **Module 8** Implementing EJB 3.0 Session Beans
- > **Module 9** The Java Persistence API
- > **Module 10** Implementing a Transaction Policy
- > **Module 11** Developing Asynchronous Java EE Applications and Messaging
- > **Module 12** Developing Message-Driven Beans

FJ310-EE6**Developing Applications for the Java EE Platform**

- > **Module 13** Web Service Model
- > **Module 14** Implementing Java EE Web Services with JAX-WS and JAX-RS
- > **Module 15** Implementing a Security Policy

SL-314-EE5**Web Component Development *with* Servlet and JSP Technologies**

- > Before: SL-275-SE6, FJ-310-EE5 After: DTJ-3108
- > Certification: Sun Certified Web Component Developer (SCWCD)
- > **Module 1** Introduction to Web Application Technologies
- > **Module 2** Developing a View Component
- > **Module 3** Developing a Controller Component
- > **Module 4** Developing Dynamic Forms
- > **Module 5** Sharing Application Resources Using the Servlet Context
- > **Module 6** Designing the Business Tier
- > **Module 7** Developing a Web Application Using Struts
- > **Module 8** Developing Web Applications Using Session Management
- > **Module 9** Using Filters in Web Applications

SL-314-EE5**Web Component Development *with* Servlet and JSP Technologies**

- > **Module 10** Integrating Web Applications With Databases
- > **Module 11** Developing JSP Pages
- > **Module 12** Developing JSP Pages Using Custom Tags
- > **Module 13** Developing Web Applications Using Struts Action Forms
- > **Module 14** Building Reusable Web Presentation Components

SL-314-EE6**Web Component Development with Servlets & JSPs, Java EE 6**

- > Before: SL-275-SE6, FJ-310-EE6 After: DTJ-3108
- > Certification: Sun Certified Web Component Developer (SCWCD)
- > **Module 1** Introduction to Java Servlets
- > **Module 2** Introduction to Java Server Pages
- > **Module 3** Implementing an MVC Design
- > **Module 4** The servlet's environment
- > **Module 5** Container facilities for servlets and JSPs
- > **Module 6** More view facilities
- > **Module 7** Developing JSP pages
- > **Module 8** Developing JSP pages using custom tags
- > **Module 9** More Controller facilities
- > **Module 10** More options for the Model

SL-314-EE6**Web Component Development with Servlets & JSPs, Java EE 6**

- > **Module 11** Asynchronous web applications
- > **Module 12** Web application security

SL-351-EE5**Business Component Development with Enterprise JavaBeans Technology**

- > Before: SL-275-SE6, FJ-310-EE5 After: SL-425, SL-500-EE5
- > Certification: Sun Certified Developer for Java Web Services
- > **Module 1** Examining EJB Applications
- > **Module 2** Introducing the Auction Application
- > **Module 3** Implementing EJB 3.0 Session Beans
- > **Module 4** Implementing Entity Classes: The Basics
- > **Module 5** Implementing Entity Classes: Modelling Data Association Relationships
- > **Module 6** Implementing Entity Classes: Modelling Inheritance Relationships
- > **Module 7** Using the Java Persistence Query Language (QL)
- > **Module 8** Developing Java EE Applications Using Messaging

SL-351-EE5**Business Component Development with Enterprise JavaBeans Technology**

- > **Module 9** Developing Message-Driven Beans
- > **Module 10** Implementing Interceptor Classes and Methods
- > **Module 11** Implementing Transactions
- > **Module 12** Handling Exceptions
- > **Module 13** Using Timer Services
- > **Module 14** Implementing Security
- > **Module 15** Using EJB Technology Best Practices

SL-351-EE6**Business Component Development with EJB Technology, Java EE 6**

Before: SL-275-SE6, FJ-310-EE6 After: SL-425, SL-500-EE5

Certification: Sun Certified Business Component Developer

- > [Module 1](#) Introduction Java EE
- > [Module 3](#) Implementing Session Beans
- > [Module 4](#) Accessing Session Beans
- > [Module 5](#) Advanced Session Bean Concepts
- > [Module 6](#) Singleton Session Bean
- > [Module 7](#) Developing Java EE Applications Using Messaging
- > [Module 8](#) Developing Message-Driven Beans
- > [Module 9](#) Using Timer Services
- > [Module 10](#) Implementing Interceptor Classes and Methods

SL-351-EE6**Business Component Development with EJB Technology, Java EE 6**

- > [Module 11](#) Implementing Transactions
- > [Module 12](#) Implementing Security
- > [Module 13](#) Using EJB Technology Best Practices

DWS-3111-EE5**Creating Web Services Using Java Technology**

Before: OO-226, FJ-310-EE5

Certification: Sun Certified Developer for Java Web Services

- > [Module 1](#) Identifying the Building Blocks of Web Services
- > [Module 2](#) Analyzing the Java Web Services Technology and Platform
- > [Module 3](#) Applying XML Processing APIs
- > [Module 4](#) Examining SOAP Messages
- > [Module 5](#) Developing Web Services Using the SOAP With Attachments API for Java (SAAJ)
- > [Module 6](#) Explaining the Web Services Description Language
- > [Module 7](#) Recognizing the Role of Service Registry
- > [Module 8](#) Implementing Web Services Using Java API for XML Web Services (JAX-WS) Technology
- > [Module 9](#) Developing Web Service Clients

**Developing Web Services Using Java Technology,
Java EE 6**

Before: FJ-310-EE6

Certification: Sun Certified Developer for Java Web Services

- > [Module 1](#) Introduction to Web Services
- > [Module 2](#) JAX-WS-based Web Services
- > [Module 3](#) SOAP and WSDL
- > [Module 4](#) JAX-WS and Java EE
- > [Module 5](#) Implementing More Complex Web Services Using JAX-WS
- > [Module 6](#) JAX-WS Web Service Clients
- > [Module 7](#) Introduction to RESTful Web Services
- > [Module 8](#) JAX-RS-based Web Services
- > [Module 9](#) JAX-RS-based Web Service Clients
- > [Module 10](#) JAX-RS and Java EE

Developing Web Services Using Java Technology, Java EE 6

- > [Module 11](#) Implementing More Complex Web Services Using JAX-RS and Jersey
- > [Module 12](#) Trade-Offs Associated with the Java Web Services APIs
- > [Module 13](#) Web Services Design Patterns
- > [Module 14](#) Best Practices and Design Patterns for Use with JAX-WS
- > [Module 15](#) Best Practices and Design Patterns for Use with JAX-RS

Developing Secure Java Web Services, Java EE 6

Before: FJ-310-EE5,

Developing Web Services using Java Tech., Java EE 6

- > [Module 1](#) Encapsulating the Basics of Security
- > [Module 2](#) Examining Web Services Security Threats and Countermeasures
- > [Module 3](#) Securing Java Web Services Using JavaEE
- > [Module 4](#) Introduction to Web Services Security
- > [Module 5](#) Web Services Security with JAX-WS and Project Metro
- > [Module 6](#) Authentication in JAX-WS
- > [Module 7](#) Identity Management and OpenSSO

SL-425

Developing Architectures for Enterprise Java Applications

- > Before: SL-314-EE5, SL-351-EE5
- > [Module 1](#) Introducing Fundamental Architectural Concepts
- > [Module 2](#) Understanding Systemic Qualities
- > [Module 3](#) Examining System Architecture Development Heuristics and Guidelines
- > [Module 4](#) Developing an Architecture for the Client Tier
- > [Module 5](#) Developing an Architecture for the Web Tier
- > [Module 6](#) Developing an Architecture for the Business Tier
- > [Module 7](#) Developing an Architecture for the Integration and Resource Tiers
- > [Module 8](#) Developing a Security Architecture
- > [Module 9](#) Evaluating the Software Architecture

SL-500-EE5

J2EE Patterns

- > Before: SL-314-EE5, SL-351-EE5, FJ-310-EE5, SL-425
- > [Module 1](#) Exploring Object-Oriented Design Principles and Design Patterns
- > [Module 2](#) Using Gang of Four Behavioral Patterns
- > [Module 3](#) Using Gang of Four Creational Patterns
- > [Module 4](#) Using Gang of Four Structural Patterns
- > [Module 5](#) Using Architectural Building Blocks
- > [Module 6](#) Introducing J2EE Patterns
- > [Module 7](#) Using Integration Tier Patterns
- > [Module 8](#) Using Presentation-to-Business Tier Patterns
- > [Module 9](#) Using Intra-Business Tier Patterns
- > [Module 10](#) Using Presentation Tier Patterns
- > [Module 11](#) More Presentation Tier Patterns

SL-500-EE5**J2EE Patterns**

- > [Module 12](#) Exploring AntiPatterns
- > [Module 13](#) Applying J2EE BluePrints Design Guidelines

DTJ-3108**Developing JavaServer Faces Components with Ajax**

Before: SL-314-EE5

- > [Module 1](#) Introducing JavaServer Faces (JSF)
- > [Module 2](#) JSF Input Components
- > [Module 3](#) Building Custom JSF Components
- > [Module 4](#) Implementing Ajax Technology
- > [Module 5](#) Ajax-Enabled JSF Components

SL-340-EE6**Developing Web Applications using JSF 2.0 Technologies**

Before: SL-275-SE6, FJ-310-EE6

- > **Module 1** Getting Started with JavaServer Faces
- > **Module 2** Designing JSF Pages Using Facelets
- > **Module 3** Using Managed Beans
- > **Module 4** Designing Navigations
- > **Module 5** Validating and Converting Data
- > **Module 6** Developing Complex Facelet Pages
- > **Module 7** Developing Composite UI Components
- > **Module 8** Working with Events
- > **Module 9** Configuring and Deploying JSF Applications

SL-370-EE6**Building Database Driven Applications with JPA**

Before: SL-275-SE6, FJ-310-EE6

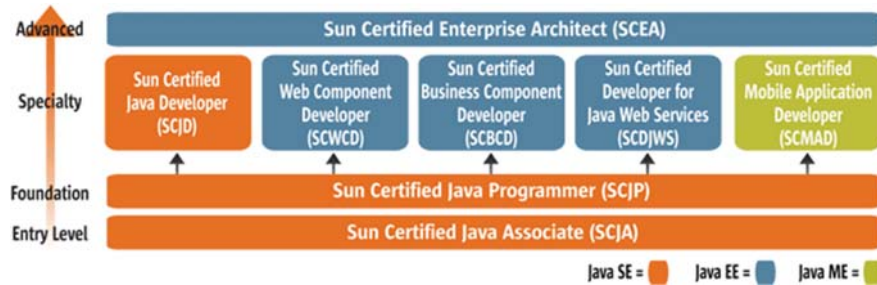
- > **Module 1** Overview of the Java Persistence API
- > **Module 2** Introducing the Auction Application
- > **Module 3** Java Persistence API Entities
- > **Module 4** Understanding the Entity Manager
- > **Module 5** Modeling Entity Relationships
- > **Module 6** Entity Inheritance and Object-Relational Mapping
- > **Module 7** Persisting Enums and Collections
- > **Module 8** Introduction to Querying
- > **Module 9** Using the Java Persistence API Query Language
- > **Module 10** Using the Java Persistence API Criteria API
- > **Module 11** Using the Java Persistence API in a Container
- > **Module 11** Implementing Transactions and Locking
- > **Module 11** Advanced Java Persistence API Concepts

3 OCP Java Certifications

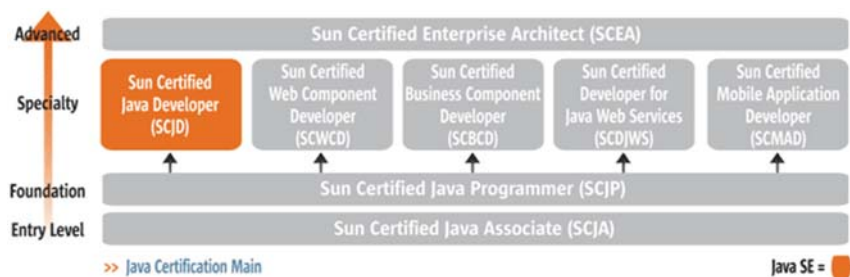
Certification Path

- > Oracle offers the following Java technology professional certifications:
 - Oracle Certified Associate, Java SE6
 - Oracle Certified Professional Java Programmer
 - Oracle Certified Master, Java SE 6 Developer
 - Oracle Certified Professional Web Component Developer
 - Oracle Certified Professional Business Component Developer
 - Oracle Certified Expert Java EE5 Web Services Developer
 - Oracle Certified Expert Java EE6 Java Persistence API Developer
 - Oracle Certified Professional Java ME 1 Mobile Application Developer
 - Oracle Certified Master, Java EE5 Enterprise Architect

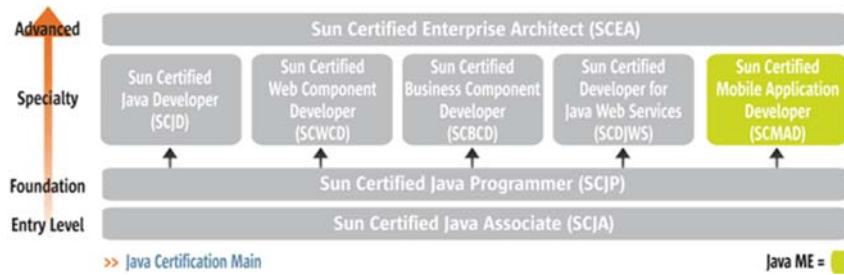
Certification Path



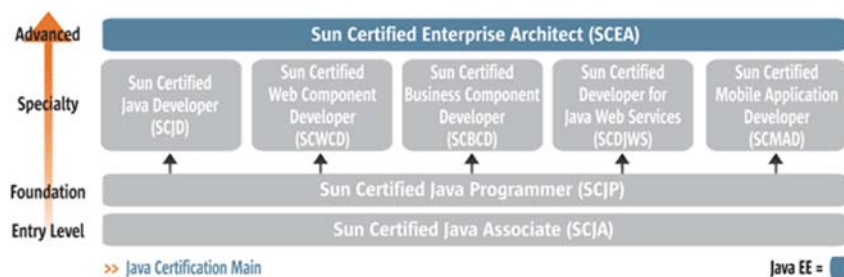
Certification Path – Java Platform SE



Certification Path – Java Platform ME



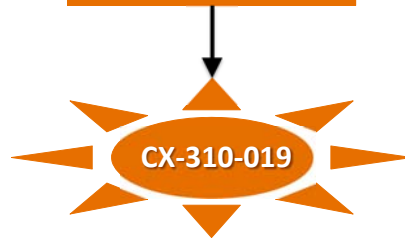
Certification Path – Java Platform EE



Oracle Certified Java Associate

Non-programmer Entry

SL-110-SE6
Fundamentals
of the Java
Programming
Language

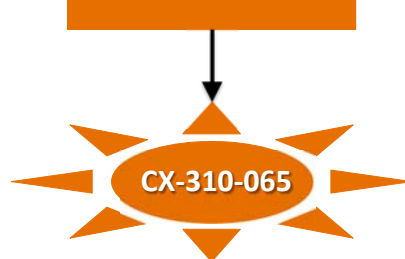


*Become
Oracle Certified
Java Associate*

Oracle Certified Professional **Java Programmer**

Experienced programmer Entry

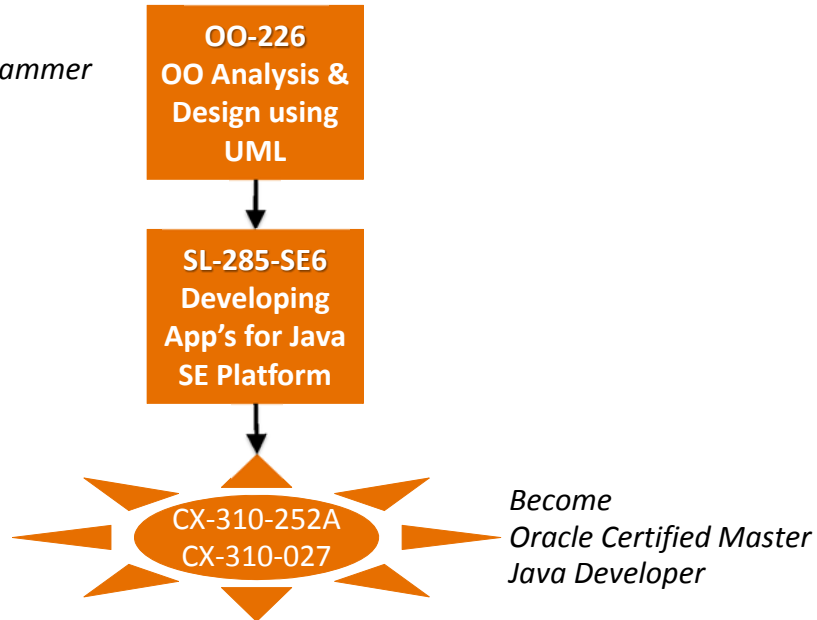
SL-275-SE6
Java
Programming
Language



*Become
Oracle Certified
Java Programmer*

Oracle Certified Master **Java Developer**

*OCJ
Java Programmer
Required*

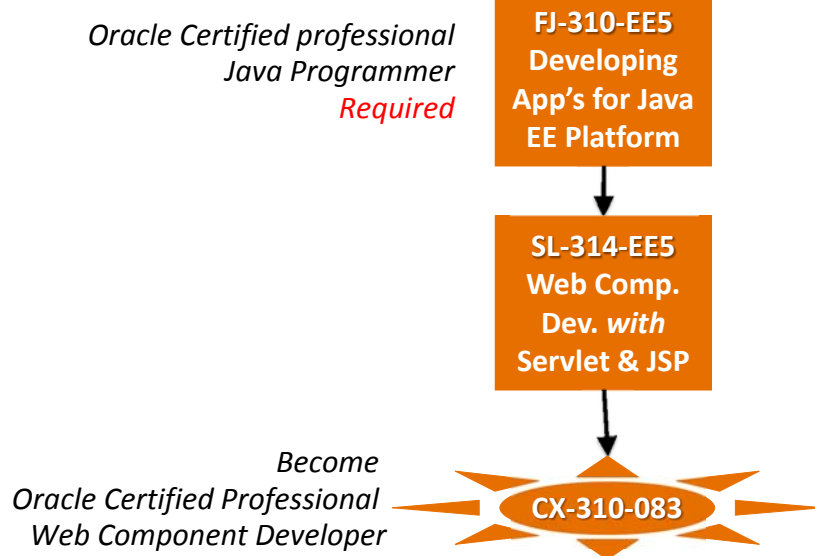


Oracle Certified Professional **Mobile Application Developer**

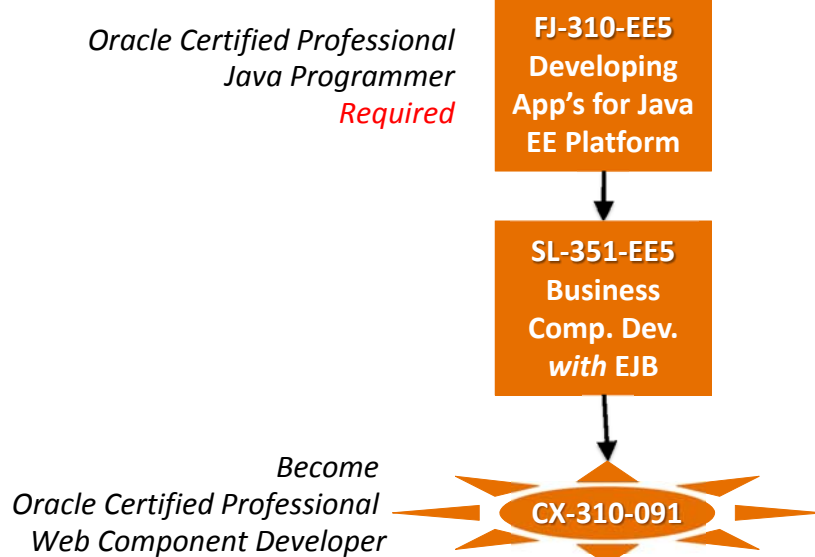
Sun Certified Java Programmer Required



Oracle Certified Professional Web Component Developer



Oracle Certified Professional Business Component Developer



Oracle Certified Expert Java EE 5 Web Services Developer

*Oracle Certified Professional
Java Programmer
Required*

FJ-310-EE5
Developing
App's for Java
EE Platform

DWS-3111-EE5
Creating Web
Services Using
Java Tech.

*Become
Oracle Certified Expert
Java EE 5 Web Services Developer*



Oracle Certified Master Java EE 5 Enterprise Architect

OO-226
OO Analysis
and Design
using UML

FJ-310-EE5
Developing
App's for Java
EE Platform

SL-425
Developing
Arch's for E.
Java App's

*Become
Oracle Certified Master
Java EE 5 Enterprise Architect*

