

Document Generated: 07/27/2024

Learning Style: Virtual Classroom

Provider: Java

Difficulty: Beginner

Course Duration: 5 Days

## Java Web Development with Servlets, JSPs, and JSF (TT5140)



## About this course:

New to Java and need to get a handle on JEE 7 basics? Need to learn Java, without the GUI and Swing? This five-day, comprehensive hands-on training course will teach you essential Java programming skills, in preparation for building server-side applications on the JEE platform. This is a five-day, comprehensive hands-on workshop geared for developers who have prior working knowledge of object-oriented programming languages such as C++. This course is geared for software developers who need to understand what JEE is, what it means in terms of today's systems and architectures, and how to apply JEE technologies and practices in implementing and deploying robust, flexible web applications. In addition to the normal exercises that are liberally sprinkled throughout the course, there is a case study that covers the entire spectrum from use cases to object-oriented design to implemented classes. This case study supplements the course and can be used during and after the course as a reference and a tool for reviewing and practicing what was learned in class.

The average salary of a Java Developer is **\$90,992** per year.

## Course Objective:

Students who attend **Java Web Developer for OO Programmers (JEE 7)** will leave the course armed with the skills they require to develop solid object-oriented applications written in Java, using sound coding techniques and best coding practices, with a specific focus on server-side development.

Java Enterprise Edition (JEE) is a powerful platform for building web applications. The JEE platform offers all the advantages of developing in Java plus a comprehensive suite of server-side technologies. This course tells you what you need to know to design and build your own web applications. You'll learn the details of the key JEE technologies and how to leverage the strengths of each, with special focus on Servlets and JSPs.

At the same time, you'll be learning about the big picture of JEE and how to design web applications that are robust, efficient, secure, and maintainable. If you want to deliver an application on the web, you'll find this course essential.

Students who attend this course will leave the course armed with the required skills to build basic web applications.

Working in a dynamic, interactive discussion and hands-on programming environment, developers will:

- Understand not only the fundamentals of the Java language, but also, it's importance, uses, strengths and weaknesses
- Understand the basics of the Java language and how it relates to OO programming and the Object Model
- Learn to use Java exception handling and logging features

- Understand and use classes, inheritance and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations
- Use the JDBC API for database access
- Work with annotations
- Design and build robust and maintainable web applications
- Create dynamic HTML content with Servlets and Java Server Pages
- Make Servlets and JSP work together cleanly

## **Audience:**

This is an introductory level training course, designed for experienced OO developers (such as C++ or SmallTalk) that need to extend their skills in web development.

## **Prerequisite:**

Understand the basic architecture of a JEE web application and/or implement a web application, taking advantage of what JEE brings to the table.

## **Course Outline:**

### **Part One: Essentials Java Programming**

#### **Module 1: Java: A First Look**

##### **Lesson: The Java Platform**

- Java Platforms
- Lifecycle of a Java Program
- Responsibilities of JVM
- Documentation and Code Reuse

##### **Lesson: Using the JDK**

- Setting Up Environment
- Locating Class Files
- Compiling Package Classes
- Source and Class Files
- Java Applications

#### **Module 2: Getting Started with Java**

##### **Lesson: Writing a Simple Class**

- Classes in Java
- Class Modifiers and Types

- Class Instance Variables
- Primitives vs. Object References
- Creating Objects

### **Lesson: Adding Methods to the Class**

- Passing Parameters into Methods
- Returning a Value from a Method
- Overloaded Methods
- Constructors
- Optimizing Constructor Usage

### **Lesson: Language Statements**

- Operators
- Comparison and Logical Operators
- Looping
- Continue and Break Statements
- The switch Statement
- The for-each() Loop

### **Lesson: Using Strings**

- Strings
- String Methods
- String Equality
- StringBuffer
- StringBuilder

### **Lesson: Specializing in a Subclass**

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining

## **Module 2: Essential Java Programming**

### **Lesson: Fields and Variables**

- Instance vs. Local Variables: Usage Differences
- Data Types
- Default Values
- Block Scoping Rules
- Final and Static Fields
- Static Methods

### **Lesson: Using Arrays**

- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments

## **Lesson: Java Packages and Visibility**

- Class Location of Packages
- The Package Keyword
- Importing Classes
- Executing Programs
- Java Naming Conventions

## **Module 3: Advanced Java Programming**

### **Lesson: Inheritance and Polymorphism**

- Polymorphism: The Subclasses
- Upcasting vs. Downcasting
- Calling Superclass Methods from Subclass
- The final Keyword

### **Lesson: Interfaces and Abstract Classes**

- Separating Capability from Implementation
- Abstract Classes
- Implementing an Interface
- Abstract Classes vs. Interfaces

### **Lesson: Exceptions**

- Exception Architecture
- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Throwing Exceptions
- Checked vs. Unchecked Exceptions

## **Module 4: Java Developer's Toolbox**

### **Lesson: Utility Classes**

- Wrapper Classes
- The Number Class
- Random Numbers
- Autoboxing/Unboxing
- The Date Class

## **Lesson: Enumerations and Static Imports**

- Enumeration Syntax
- When You Should Use Enumerations
- Using Static Imports
- When You Should Use Static Imports

## **Module 5: Collections and Generics**

### **Lesson: Introduction to Generics**

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls to Generics
- When Generics Should Be Used

### **Lesson: Collections**

- Characterizing Collections
- Collection Interface Hierarchy
- Iterators
- The Set Interface
- The List Interface
- Queue Interface
- Map Interfaces
- Using the Right Collection
- Collections and Multithreading

## **Module 6: Java Application Development**

### **Lesson: Introduction to Annotations**

- Annotations Overview
- Working with Java Annotations

## **Part 2: Session: Developing Java EE Applications**

## **Module 7: Developing Java EE applications**

### **Lesson: Enterprise Development**

- Enterprise Application Software
- Requirements of Enterprise applications
- Scalability, Load Balancing, Fail Over
- Resource pooling

## **Lesson: Java EE Core Components**

- Overview of Java EE Core Components
- Web Tier Components
- Application Tier
- Deployable Units
- Deployment Descriptors
- The Java Naming and Directory Interface (JNDI)

## **Module 8: JEE Dynamic Web Applications**

### **Lesson: Introduction to Servlets**

- The Servlet Interface
- The Web Container
- Creating HTML Output Using Servlets
- The @WebServlet Annotation
- Interaction Between web.xml and Annotations
- The @WebInitParam Annotation

### **Lesson: Form processing using Servlets**

- Using HTML5 Forms with Servlets
- Processing Request Parameters
- HttpServletRequest Methods
- HttpServletResponse Methods

### **Lesson: Java Server Pages**

- Java Server Pages (JSPs)
- The Relationship Between JSPs and Servlets
- The JSP lifecycle
- The role of JSPs in Java EE 7

### **Lesson: Implementing MVC in JEE**

- Model View Control
- Using the RequestDispatcher
- Handling Requests
- The Request Scope
- Handling Request Attributes
- The Expression Language (JSR 341)
- EL in Template text

### **Lesson: Session Management**

- Sessions in Web Applications
- The HttpSession object
- Session Management in Java EE

- Handling Cookies
- URL-Rewriting

## **Module 9: Using Resources**

### **Lesson: Java Data Access JDBC API**

- Connecting to the Database
- Statement and PreparedStatement
- ResultSet
- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency

### **Lesson: JEE DataSources**

- DataSources in JEE
- Setup a DataSource
- Using CDI to inject a DataSource

### **Lesson: Overview of JPA**

- Introduce the Java Persistence API (JPA)
- Benefits of Using an ORM framework
- Hibernate and JPA

## **Module 10: Advanced Topics**

### **Lesson: XML Overview**

- XML Separates Structure, Content and Format
- Content: XML Document Syntax Rules
- Structure: A Document Type Definition
- Formatting: XSLT Transformation to HTML

### **Lesson: Webservices Overview**

- Web Services Characteristics
- Web Services Architecturally
- Web Services Challenges
- .NET Web Services
- Java and Web Services
- Web Services in Action

### **Lesson: Overview of REST**

- REpresentational State Transfer
- REST Characteristics
- REST Elements



- REST Architectural Principles
- REST and HTTP
- REST/HTTP: Representation-Oriented
- REST Design Principles

## Credly Badge:



### Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your LinkedIn profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

[Find Out More](#) or [See List Of Badges](#)