Java Advanced (Programming SE 8 – OCP)

Prerequisites / Further Training

- Java Beginner
- Beginning SQL

Also have a look at our Java Bootcamp

Alignment

Oracle OCP Certification aligned to Oracle OCP Java Exam

After this course you should be able to

- Have a good understanding of programming and the building blocks of an OO programming language, with an emphasis on JAVA.
- Build small apps in Java, making use of I/O, Networking, GUI
- Prepare for Oracle OCA and OCP exams

Course Material

Course Material Provided

Course Contents

DAY 1

Abstract and Nested Classes

- Modeling Business Problems with Classes
- Enabling Generalization
- Identifying the Need for Abstract Classes
- Defining Abstract Classes
- Defining Abstract Methods
- Validating Abstract Classes

- Final Methods
- Final Classes
- Final Variables
- Declaring Final Variables
- Nested Classes
- Example: Member Class
- Enumerations
- Enum Usage
- Complex Enums

Interfaces and Lambda Expressions

- Java Interfaces
- A Problem Solved by Interfaces
- CrushedRock Class
- The SalesCalcs Interface
- Adding an Interface
- Interface References
- Interface Reference Usefulness
- Interface Code Flexibility
- default Methods in Interfaces
- default Method: Example
- static Methods in Interfaces
- Constant Fields
- Extending Interfaces
- Implementing and Extending
- Anonymous Inner Classes
- Anonymous Inner Class: Example
- String Analysis Regular Class
- String Analysis Regular Test Class
- String Analysis Interface: Example
- String Analyzer Interface Test Class
- Encapsulate the for Loop
- String Analysis Test Class with Helper Method
- String Analysis Anonymous Inner Class
- String Analysis Lambda Expression
- Lambda Expression Defined

- What Is a Lambda Expression?
- Lambda Expression Shorthand
- Lambda Expressions as Variables

7 Generics and Collections

- Generics 7-4
- Simple Cache Class Without Generics 7-5
- Generic Cache Class 7-6
- Generics in Action 7-7
- Generics with Type Inference Diamond 7-8
- Collections 7-9
- Collection Types 7-10
- Collection Interfaces and Implementation 7-11
- List Interface 7-12
- ArrayList 7-13
- Autoboxing and Unboxing 7-14
- ArrayList Without Generics 7-15
- Generic ArrayList 7-16
- Generic ArrayList: Iteration and Boxing 7-17
- Set Interface 7-18
- TreeSet: Implementation of Set 7-19
- Map Interface 7-20
- Map Types 7-21
- TreeMap: Implementation of Map 7-22
- Deque Interface 7-23
- Stack with Deque: Example 7-24
- Ordering Collections 7-25

DAY 2

Collections, Streams, and Filters

- Collections, Streams, and Filters 8-3
- The Person Class 8-4
- Person Properties 8-5
- Builder Pattern 8-6
- Collection Iteration and Lambdas 8-7

- RoboCallTest07: Stream and Filter 8-8
- RobocallTest08: Stream and Filter Again 8-9
- SalesTxn Class 8-10
- Java Streams 8-11
- The Filter Method 8-12
- Method References 8-13
- Method Chaining 8-14
- Pipeline Defined 8-16

Lambda Built-in Functional Interfaces

- Built-in Functional Interfaces 9-3
- The java.util.function Package 9-4
- Example Assumptions 9-5
- Predicate 9-6
- Predicate: Example 9-7
- Consumer 9-8
- Consumer: Example 9-9
- Function 9-10
- Function: Example 9-11
- Supplier 9-12
- Supplier: Example 9-13
- Primitive Interface 9-14
- Return a Primitive Type 9-15
- Return a Primitive Type: Example 9-16
- Process a Primitive Type 9-17
- Process Primitive Type: Example 9-18
- Binary Types 9-19
- Binary Type: Example 9-20
- Unary Operator 9-21
- UnaryOperator: Example 9-22
- Wildcard Generics Review 9-23

Lambda Operations

- Objectives 10-2
- Streams API 10-3
- Types of Operations 10-4

- Extracting Data with Map 10-5
- Taking a Peek 10-6
- Search Methods: Overview 10-7
- Search Methods 10-8
- Optional Class 10-9
- Lazy Operations 10-10
- Stream Data Methods 10-11
- Performing Calculations 10-12
- Sorting 10-13
- Comparator Updates 10-14
- Saving Data from a Stream 10-15
- Collectors Class 10-16
- Quick Streams with Stream.of 10-17
- Flatten Data with flatMap 10-18

DAY 3

Exceptions and Assertions

- Error Handling 11-3
- Exception Handling in Java 11-4
- try-catch Statement 11-5
- Exception Objects 11-6
- Exception Categories 11-7
- Handling Exceptions 11-8
- finally Clause 11-9
- try-with-resources Statement 11-10
- Catching Multiple Exceptions 11-11
- Declaring Exceptions 11-12
- Handling Declared Exceptions 11-13
- Throwing Exceptions 11-14
- Custom Exceptions 11-15
- Assertions 11-16
- Assertion Syntax 11-17
- Internal Invariants 11-18
- Control Flow Invariants 11-19
- Class Invariants 11-20

Java Date/Time API

Java I/O Fundamentals

- Byte Stream: Example 13-9
- Character Stream Reader Methods 13-10
- Character Stream Writer Methods 13-11
- Character Stream: Example 13-12
- I/O Stream Chaining 13-13
- Chained Streams: Example 13-14
- Console I/O 13-15
- Writing to Standard Output 13-16
- Reading from Standard Input 13-17
- Channel I/O 13-18
- Persistence 13-19
- Serialization and Object Graphs 13-20
- Transient Fields and Objects 13-21
- Transient: Example 13-22
- Serial Version UID 13-23
- Serialization: Example 13-24
- Writing and Reading an Object Stream 13-25
- Serialization Methods 13-26

DAY 4

Java File I/O (NIO.2)

- Objectives 14-2
- New File I/O API (NIO.2) 14-3
- Limitations of java.io.File 14-4
- File Systems, Paths, Files 14-5
- Relative Path Versus Absolute Path 14-6
- Java NIO.2 Concepts 14-7
- Path Interface 14-8
- Path Interface Features 14-9
- Path: Example 14-10
- Removing Redundancies from a Path 14-11
- Creating a Subpath 14-12
- Joining Two Paths 14-13
- Symbolic Links 14-14
- Working with Links 14-15

- Styles Compared 17-7
- Parallel Stream 17-8
- Using Parallel Streams: Collection 17-9
- Using Parallel Streams: From a Stream 17-10
- Pipelines Fine Print 17-11
- Embrace Statelessness 17-12
- Avoid Statefulness 17-13
- Streams Are Deterministic for Most Part 17-14
- Some Are Not Deterministic 17-15
- Reduction 17-16
- Reduction Fine Print 17-17
- Reduction: Example 17-18
- A Look Under the Hood 17-24
- Illustrating Parallel Execution 17-25
- Performance 17-36
- A Simple Performance Model 17-37

DAY 5

Building Database Applications with JDBC

- Objectives 18-2
- Using the JDBC API 18-3
- Using a Vendor’s Driver Class 18-4
- Key JDBC API Components 18-5
- Writing Queries and Getting Results 18-6
- Using a ResultSet Object 18-7
- CRUD Operations Using JDBC API: Retrieve 18-8
- CRUD Operations Using JDBC: Retrieve 18-9
- CRUD Operations Using JDBC API: Create 18-10
- CRUD Operations Using JDBC API: Update 18-11
- CRUD Operations Using JDBC API: Delete 18-12
- SQLException Class 18-13
- Closing JDBC Objects 18-14
- try-with-resources Construct 18-15
- Using PreparedStatement 18-16
- Using PreparedStatement: Setting Parameters 18-17

- Executing PreparedStatement 18-18
- PreparedStatement:Using a Loop to Set Values 18-19
- Using CallableStatement 18-20

Localization

- Objectives 19-2
- Why Localize? 19-3
- A Sample Application 19-4
- Locale 19-5
- Properties 19-6
- Loading and Using a Properties File 19-7
- Loading Properties from the Command Line 19-8
- Resource Bundle 19-9
- Resource Bundle File 19-10
- Sample Resource Bundle Files 19-11
- Initializing the Sample Application 19-12
- Sample Application: Main Loop 19-13
- The printMenu Method 19-14
- Changing the Locale 19-15
- Sample Interface with French 19-16
- Format Date and Currency 19-17
- Displaying Currency 19-18
- Formatting Currency with NumberFormat 19-19
- Displaying Dates 19-20
- Displaying Dates with DateTimeFormatter 19-21
- Format Styles 19-22

Duration and pricing

Price Group A

Certificate

Read About Our Certificates

Bookings

You can download the course registration form on our home page

or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please email us

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

Java Beginner (SE 8 Fundamentals – OCA)

This Beginner Java Training Course will you the fundamentals of the Java Programming Language with an emphasis on OO. This course is aligned with the Oracle Certified Associate in Java Programming Certification.

Prerequisites / Further Training

You should not be a complete beginner for this course. If you cannot pass this test, you must do Intro To Programming Course first.

Alignment

OCA: Oracle Certified Associate Java SE 8 Programmer Exam 1Z0-808

Intended Audience

- Intended for people who has some knowledge of programming and wanting to learn Java and OO
- **NB**: This is NOT an Introduction to Programming course.

After this course you should be able to

- Have a good understanding of programming and the building blocks of an OO programming language, with an emphasis on JAVA. Prepare for Oracle OCA Exam 1Z0-803
- Use Java programming language constructs to create a Java technology application.
- Use decision and looping constructs and methods to dictate program flow.
- Understand basic object oriented concepts such as inheritance, encapsulation, and abstraction.
- Use and manipulate object references, and to write simple error handling code.
- Use the new SE 8 java.time and java.time.format packages to format and print the local date and time.
- Specify a data modification by passing a predicate lambda expression to the Collections class.
- Proceed to practical training that assume OO knowledge like Advanced Java, Angular, React and more

Course Material

Course Material Provided

Course Contents
Day 1
Introducing Java Technology

- Breaking the Surface
- The way Java works

- Code Structure in Java
- Anatomy of a class
- The main() method
- Netbeans IDE and Debugging

Loop and decision constructs

- Looping
- Conditional branching
- A trip to Objectville
- Inheritance,Overriding
- Class variables and methods
- Making your first object,Using main

Day 2

Primitives

- Know your Variables
- Declaring a variable
- Primitive types
- Java keywords

Objects

- Reference variables
- Object declaration and assignment
- Objects on the garbage collectible heap
- Arrays
- How Objects Behave
- Methods use object state
- Method arguments and return types
- Pass-by-value

Encapsulation

- Getters and Setters
- Encapsulation
- Using references in an array

Day 3

Arrays and Arraylists

- Extra Strength Methods
- Building a one-dim Arraylist game
- Preparing to code
- Coding
- Random numbers<
- Using user-input
- For loops
- Casting primitives
- String conversion
- Using the Java Library
- Two Dimensional Arraylist Structures
- Enhancing the game
- Coding the game
- Boolean expressions
- Using the Java library (API)
- Using packages
- Using the HTML API docs and

Day 4

Polymorphism. Method Overloading

- Better Living in Objectville
- Understanding inheritance
- Designing an inheritance tree
- Avoiding duplicate code
- Overriding methods
- IS-A and HAS-A · What do you inherit from your superclass?
- What does inheritance really buy you?
- Polymorphism
- Rules for overriding
- Method overloading

Advanced OO Concepts

- Serious Polymorphism
- Some classes should not be instantiated
- Abstract classes
- Abstract methods
- Polymorphism in action
- Class Object
- Taking objects out of an arraylist
- Compiler checks the reference type
- Get in touch with your inner object
- Polymorphic references
- Casting an object reference (moving lower in the inheritance tree)
- Deadly Diamond of Death
- Using interfaces (the best solution

Day 5

Garbage Collection

- Life and Death of an Object
- The stack and the heap
- Methods on the stack
- Where local variables live
- Where instance variables live
- The miracle of object creation

Constructors

- Constructors, Initializing state of a new Object
- Overloaded constructors
- Superclass constructors
- Invoking overloaded constructors using this()
- Life of an object, Garbage collection

Handling Errors

- Handling Errors
- Handling Exceptions

Duration and pricing

In Price Group A

Certificate.

Please read about our ***certificates***.

Bookings

You can download the course registration form on our home page or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please email us

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

Java Enterprise Architect

Prerequisites

You should be at the level of our Java Spring / EE Bootcamp plus the following:

Design Patterns

OO and UML

EJB Fundamentals

Java Enterprise Architect

Intended Audience

- This course is Intended for developers who are fluent in at least Java Spring Web Applications as per our Java Developer Bootcamp and is NOT an introductory course

After this course you should be able to

- Have all the skills required to start preparing for the Oracle Java EJB and Enterprise Architect Exams

Course Material

- Course Material Provided

Course Contents

Day 1

- The Oracle Certified Master, Java EE 6 Enterprise Architect Exam and Certification
- Application Design Concepts and Principles

Day 2

- Common Architectures
- Integration and Messaging

Day 3

- Business Tier Technologies
- Web Tier Technologies

Day 4

- Design Patterns
- Security

Day 5

- Java (EE) Enterprise Architect Certified Master Assignment
- Java (EE) Enterprise Architect Certified Master Essay Exam

Duration and pricing

Price Group A

Certificate

Read About Our Certificates

Schedule

On the calender on this page below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our 'Contact Us' page.

Bookings

You can download the course registration form on our home page or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the bottom of the page.

Questions

Please email us

Java OCA Certification Exam Workshop

Java OCEJWCD Certification Workshop

Java OCP Certification Exam Workshop

Alignment

Aligned to Oracle OCP Certification aligned to Oracle OCP Java Exam, however this course is recommended as an exam prep. You should not attemt to study for the OCP eam before you master the practical skills taught in this course.

After this course you should be able to

- Have a good understanding of programming and the building blocks of an OO programming language, with an emphasis on JAVA.
- Build small apps in Java, making use of I/O, Networking,

GUI
- Prepare for Oracle OCA and OCP exams

Course Material

Course Material Provided

Course Contents

DAY 1:

Class Design

- Encapsulation and Immutable Classes
- Inheritance and Polymorphism
- Inner Classes
- Interfaces
- Enumerations

Generics and Collections

- Generics
- Collections

DAY 2:

Lambda Expressions

- Functional Interfaces
- Lambda Expressions
- Java Built-In Lambda Interfaces
- Method References

Streams and Collections

- Streams
- Iterating and Filtering Collections
- Optional Class
- Data Search
- Stream Operations on Collections
- Parallel Streams
- Peeking, Mapping, Reducing and Collecting
- Files and Streams

DAY 3:

Exceptions and Assertions

- Exceptions
- Assertions

Date/Time API

- Core Date/Time Classes
- Time Zones and Daylight Savings

DAY 4:

Java I/O

- Java I/O Fundamentals
- NIO.2

Concurrency

- Thread Basics
- Concurrency
- Fork/Join Framework

DAY 5:

JDBC and Localization

- JDBC API
- Localization

Duration and pricing

- *Price Group A*

Certificate

Read About Our Certificates

Bookings

You can download the course registration form on our home page or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please email us

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

Java Persistence API

This Java Persistence API training course explores using the Java Persistence API within the context of a

- Java Standard Edition application
- Web-based Java Enterprise Edition application (and using Java Persistence API with EJB's).

Prerequisites / Further Training

Java Servlets Beginner	Java Servlets Advanced
Java Enterprise Edition	Enterprise Java Beans
Java Persistence API	Java Web Services

Java Server Faces (under construction)	Spring Framework

You do not need to be familiar with Java EE

Course Material

- Included in the course price.
- We use Glassfish Application Server 4.0 + for our course exercises

After this course you should be able to:

- Understand key concepts found in the Java Persistence API.
- Update multiple database tables based on relationships.
- Perform CRUD operations with JPA in Java SE and EE environments.
- Perform data validation using Bean Validation.
- Optimize JPA for performance. Apply transactions and locking.
- Map relational database tables to Java using ORM techniques and JPA.
- Create robust entity models.
- Create static and dynamic queries using JPQL.
- Create type-safe queries with the Java Persistence API Criteria API.
- Prepare for the Java EE 6 Java Persistence API Developer Certified Expert 1Z0-898 exam (not included)

Contents

Day 1

Introduction

- Relational Databases
  Object-Relational Mapping
  The Impedance Mismatch
  Java Support for Persistence

Proprietary Solutions
JDBC
Enterprise JavaBeans
Java Data Objects
Why Another Standard?
The Java Persistence API
History of the Specification

Getting Started

- Entity Overview
  Persistability
  Identity
  Transactionality
  Granularity
  Entity Metadata
  Annotations
  Configuration by Exception
  Creating an Entity
  Entity Manager
  Obtaining an Entity Manager
  Persisting an Entity
  Finding an Entity
  Removing an Entity
  Updating an Entity
  Transactions
  Queries
  Putting It All Together
  Packaging It Up
  Persistence Unit
  Persistence Archive

Enterprise Applications

- Application Component Models
  Session Beans
  Stateless Session Beans
  Stateful Session Beans

Singleton Session Beans
Servlets
Dependency Management and CDI
Dependency Lookup
Dependency Injection
Declaring Dependencies
- CDI and Contextual Injection

CDI Beans
Injection and Resolution
Scopes and Contexts
Qualified Injection
Producer Methods and Fields
Using Producer Methods with JPA Resources
Transaction Management
Transaction Review
Enterprise Transactions in Java
Putting It All Together
Defining the Component
Defining the User Interface
Packaging It Up

Day 2

Object-Relational Mapping

- Persistence Annotations

Accessing Entity State
Field Access
Property Access
Mixed Access
Mapping to a Table
Mapping Simple Types
Column Mappings
Lazy Fetching
Large Objects
Enumerated Types

Temporal Types
Transient State
Mapping the Primary Key
Overriding the Primary Key Column
Primary Key Types
Identifier Generation
Relationships
Relationship Concept
Mappings Overview
Single-Valued Associations
Collection-Valued Associations
Lazy Relationships
Embedded Objects

Collection Mapping

- Relationships and Element Collections
  Using Different Collection Types
  Sets or Collections
  Lists
  Maps
  Duplicates
  Null Values
  Best Practices

Entity Manager

- Persistence Contexts
  Entity Managers
  Container-Managed Entity Managers
  Application-Managed Entity Managers
  Transaction Management
  JTA Transaction Management
  Resource-Local Transactions
  Transaction Rollback and Entity State
  Choosing an Entity Manager
  Entity Manager Operations
  Persisting an Entity

Finding an Entity
Removing an Entity
Cascading Operations
Clearing the Persistence Context
Synchronization with the Database
Detachment and Merging
Detachment
Merging Detached Entities
Working with Detached Entities

Day 3

Using Queries

- Java Persistence Query Language
  Getting Started
  Filtering Results
  Projecting Results
  Joins Between Entities
  Aggregate Queries
  Query Parameters
  Defining Queries
  Dynamic Query Definition
  Named Query Definition
  Dynamic Named Queries
  Parameter Types
  Executing Queries
  Working with Query Results
  Query Paging
  Queries and Uncommitted Changes
  Query Timeouts
  Bulk Update and Delete
  Using Bulk Update and Delete
  Bulk Delete and Relationships
  Query Hints
  Query Best Practices

Named Queries
Report Queries
Vendor Hints
Stateless Beans
Bulk Update and Delete
Provider Differences

Query Language

- Introducing JP QL
  Terminology
  Example Data Model
  Example Application
  Select Queries
  SELECT Clause
  FROM Clause
  WHERE Clause
  Inheritance and Polymorphism
  Scalar Expressions
  ORDER BY Clause
  Aggregate Queries
  Aggregate Functions
  GROUP BY Clause
  HAVING Clause
  Update Queries
  Delete Queries

Criteria API

- Overview
  The Criteria API
  Parameterized Types
  Dynamic Queries
  Building Criteria API Queries
  Creating a Query Definition
  Basic Structure
  Criteria Objects and Mutability
  Query Roots and Path Expressions

The SELECT Clause
The FROM Clause
The WHERE Clause
Building Expressions
The ORDER BY Clause
The GROUP BY and HAVING Clauses
Bulk Update and Delete
Strongly Typed Query Definitions
The Metamodel API
Strongly Typed API Overview
The Canonical Metamodel
Choosing the Right Type of Query

Day 4

Advanced Object-Relational Mapping

- Table and Column Names
  Converting Entity State
  Creating a Converter
  Declarative Attribute Conversion
  Automatic Conversion
  Converters and Queries
  Complex Embedded Objects
  Advanced Embedded Mappings
  Overriding Embedded Relationships
  Compound Primary Keys
  Id Class
  Embedded Id Class
  Derived Identifiers
  Basic Rules for Derived Identifiers
  Shared Primary Key
  Multiple Mapped Attributes
  Using EmbeddedId
  Advanced Mapping Elements
  Read-Only Mappings

Optionality
Advanced Relationships
Using Join Tables
Avoiding Join Tables
Compound Join Columns
Orphan Removal
Mapping Relationship State
Multiple Tables
Inheritance
Class Hierarchies
Inheritance Models
Mixed Inheritance

Advanced Queries

- SQL Queries

Native Queries vs. JDBC
Defining and Executing SQL Queries
SQL Result Set Mapping
Parameter Binding
Stored Procedures
Entity Graphs
Entity Graph Annotations
Entity Graph API
Managing Entity Graphs
Using Entity Graphs

Other Advanced Topics

- Lifecycle Callbacks

Lifecycle Events
Callback Methods
Entity Listeners
Inheritance and Lifecycle Events
Validation
Using Constraints
Invoking Validation
Validation Groups

Creating New Constraints
Validation in JPA
Enabling Validation
Setting Lifecycle Validation Groups
Concurrency
Entity Operations
Entity Access
Refreshing Entity State
Locking
Optimistic Locking
Pessimistic Locking
Caching
Sorting Through the Layers
Shared Cache
Utility Classes
PersistenceUtil
PersistenceUnitUtil

Day 5

XML Mapping Files

- The Metadata Puzzle
  The Mapping File
  Disabling Annotations
  Persistence Unit Defaults
  Mapping File Defaults
  Queries and Generators
  Managed Classes and Mappings
  Converters

Packaging and Deployment

- Configuring Persistence Units
  Persistence Unit Name
  Transaction Type
  Persistence Provider

Data Source
Mapping Files
Managed Classes
Shared Cache Mode
Validation Mode
Adding Properties
Building and Deploying
Deployment Classpath
Packaging Options
Persistence Unit Scope
Outside the Server
Configuring the Persistence Unit
Specifying Properties at Runtime
System Classpath
Schema Generation
The Generation Process
Deployment Properties
Runtime Properties
Mapping Annotations Used by Schema Generation
Unique Constraints
Null Constraints
Indexes
Foreign Key Constraints
String-Based Columns
Floating Point Columns
Defining the Column

Testing

- Testing Enterprise Applications

Terminology
Testing Outside the Server
JUnit
Unit Testing
Testing Entities
Testing Entities in Components
The Entity Manager in Unit Tests

_____-

Duration and pricing

In Price Group A

Certificate

Read about our Certificates

Bookings

You can download the course registration form on our home page or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please email us Schedule On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

Java Server Faces

This Java Server Faces Training Course covers all the important aspects involved in developing JSF 2.2 applications. Get the most out of JSF 2.2 practically

Prerequisites

You should be at least on the level of the Java Servlets

Beginner course before doing this course.

Intended Audience

Java Developers who need to hone their front-end skills. This course covers all the important aspects involved in developing JSF 2.2 applications. It provides clear instructions for getting the most out of JSF 2.2 and with many exercises , honing practical skills in JSF in order to build impressive JSF-based web applications.

After this course you should be able to:

Build impressive JSF-based web applications. It will also fortify your knowledge about JSF 2.2 Facelets.

Course Contents

Day 1:

Dynamic Access to JSF Application Data through Expression Language (EL 3.0)

Communication

Day 2

JSF Scopes – Lifespan and Use in Managed Beans Communication

JSF Configurations Using XML Files and Annotations – Part 1

JSF Configurations Using XML Files and Annotations – Part 2

Day 3

Working with Tabular Data

JSF and AJAX

JSF 2.2 – HTML5 and Upload

Day 4

JSF State Management

JSF Custom Components

Day 5

JSF 2.2 Resource Library Contracts – Themes

Facelets Templating

Duration and pricing

- In Pricing Group A

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please email us

Schedule

On the calendar below. If your browser doesn’t display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

Java Web Services

Prerequisites

You should know how to develop a complete Web App like the one we do in our Spring MVC course.

Intended Audience

Java Web Developers who want to improve/consolidate their skills in web services

After this course you should be able to

- Develop REST-style and SOAP-based web services and clients with this quick and thorough introduction.
- Prepare for the Java EE 6 Web Services Developer Certified Expert 1Z0-897 exam

Course Material

Supplied in electronic format

Course Contents

Day 1

Web Services Quickstart

- Web Service Miscellany
- What Good Are Web Services?
- Web Services and Service-Oriented Architecture
- A Very Short History of Web Services
- What Is REST?
- Review of HTTP Requests and Responses
- HTTP as an API
- A First RESTful Example
- Why Use Servlets for RESTful Web Services?

Day 2

RESTful Web Services: The Service Side

- A RESTful Service as an HttpServlet
- A RESTful Web Service as a JAX-RS Resource
- A RESTful Web Service as Restlet Resources
- A RESTful Service as a @WebServiceProvider

RESTful Web Services: The Client Side

- A Perl Client Against a Java RESTful Web Service
- A Client Against the Amazon E-Commerce Service
- A Standalone JAX-B Example
- Another Client Against the Amazon E-Commerce Service
- The CTA Bus-Tracker Services
- RESTful Clients and WADL Documents
- The JAX-RS Client API
- JSON for JavaScript Clients

Day 3

SOAP-Based Web Services

- A SOAP-Based Web Service
- The RandService in Two Files
- Clients Against the RandService
- The WSDL Service Contract in Detail
- SOAP-Based Clients Against Amazon's E-Commerce Service

SOAP Handlers and Faults

- The Handler Level in SOAP-Based Services and Clients
- Handlers and Faults in the predictionsSOAP Service
- A Handler Chain with Two Handlers
- SOAP-Based Web Services and Binary Data
- The Transport Level
- Axis2

Day 4

Web Services Security

- Wire-Level Security
- A Very Lightweight HTTPS Server and Client
- HTTPS in a Production-Grade Web Server
- Container-Managed Security
- WS-Security

Day 5

Web Services and Java Application Servers

- The Web Container
- Toward a Lightweight JAS
- GlassFish Basics
- Servlet-Based Web Services Under GlassFish
- An Interactive Website and a SOAP-Based Web Service
- A @WebService as a @Stateless Session EJB
- TomEE: Tomcat with Java EE Extensions
- Where Is the Best Place to Be in Java Web Services?

Duration and pricing

- In pricing Group A

Certificate

- Please read about our certificates

Bookings

You can download the course registration form on our home page or by clicking here

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please email us Schedule On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

_____

JHipster (Spring, Angular)

This **JHipster Training Course with Spring and Angular** will teach you how to build modern web applications and microservices with Spring and Angular using JHipster, which sits in between the two and can generate templates for both the front and back-end. It is also used to control database changes, the architecture of the application and much more.

Prerequisites

- Needless to say, your Java knowledge should be up to scratch – at least on Java 8 Programming level.
- This course assumes and therefore only touches on Web API protocols. If you do not know Web Services, you must do Spring REST API to complete your full-stack.
- This course only touches on Angular. If you do not know Angular, you must do Angular to complete your full-stack.

Further Training

Spring MVC, Spring REST API, Angular, Java Web Services

Contents

Introduction to Modern Web Application Development

- Modern full-stack web development
- Web architecture patterns
- Monolithic web architecture
- Microservice architecture
- Choosing the right pattern
- When to choose a monolithic architecture
- When to choose a microservice architecture

Getting Started with JHipster

- Why JHipster?
- Goal and adoption of JHipster
- Introduction to technologies available
- Client-side technologies
- HTML5 and CSS3
- HTML5
- CSS3
- Sass
- Bootstrap
- MVVM framework
- Angular
- React

- Build tools
- Webpack
- BrowserSync
- Testing tools
- Karma
- Protractor
- Internationalization
- Server-side technologies
- Spring Framework
- Spring Boot
- Spring Security
- Spring MVC
- Spring data
- Security
- JWT
- Session
- OAuth2
- Build tools
- Maven
- Gradle
- Hibernate
- Liquibase
- Caching
- Ehcache
- Hazelcast
- Infinispan
- Swagger
- Thymeleaf
- Dropwizard metrics
- WebSocket
- Kafka
- Testing frameworks
- JUnit
- Gatling
- Cucumber
- Introduction to database options
- SQL databases

- H2
- MySQL
- MariaDB
- PostgreSQL
- MS SQL
- Oracle
- NoSQL databases
- MongoDB
- Cassandra
- Elasticsearch
- Installation and setup
- Prerequisites
- Installation procedure
- Java 8
- Git
- Node.js
- Yarn
- Docker
- IDE configuration
- System setup
- Installation of JHipster

Building Monolithic Web Applications with JHipster

- Application generation
- Step 1 – preparing the workspace
- Step 2 – generating code using JHipster
- Server-side options
- Client-side options
- Internationalization options
- Testing
- Modules
- Code walkthrough
- File structure
- Server-side source code
- Java source
- Resources

- client-side source code
- Starting the application
- Application modules
- Home and Login modules
- Account modules
- Settings
- Password
- Registration
- Admin module
- User management
- Metrics
- Health
- Configuration
- Audits
- Logs
- API
- Running generated tests
- Server-side tests
- Client-side tests

Entity Modeling with JHipster Domain Language

- Introduction to JDL
- DSL grammar for JDL
- Entity modelling with JDL
- Relationship management
- DTO, service, and pagination options
- JDL Studio
- Use case entity model with an explanation
- Entities
- Relationships
- Options for entities
- Entity generation with JHipster
- Generated code walkthrough
- Server-side source code
- Domain class for the entity
- Repository interface for the entity

- Service class for the entity
- Resource class for the entity
- Client side
- TypeScript model class for the entity
- Angular services for the entity
- Angular components of the entity
- Angular route for the entity
- Angular module for the entity
- Generated pages
- Running generated tests

Customization and Further Development

- Live reload for development
- Spring Boot DevTools
- Webpack dev server and BrowserSync
- Setting up live reload for an application
- Customizing the Angular frontend for an entity
- Editing an entity using the JHipster entity sub-generator
- Changing the look and feel of the application
- Adding a new i18n language
- Authorization with Spring Security
- Limiting access to entities
- Limiting access to create/edit/delete entities
- Limiting access to data of other users

Testing and Continuous Integration

- Fixing and running tests
- Continuous integration
- CI/CD tools
- Jenkins
- Travis CI
- GitLab CI
- CircleCI
- Setting up Jenkins
- Creating a Jenkins pipeline using JHipster

- The Jenkinsfile and its stages
- Setting up the Jenkinsfile in a Jenkins server

Going into Production

- An Introduction to Docker
- Docker containers
- The Dockerfile
- The Docker Hub
- Docker compose
- Starting the production database with Docker
- An introduction to Spring profiles
- Packaging the application for local deployment
- Building and deploying using Docker
- Building and deploying an executable archive
- Upgrading to the newest version of JHipster
- An introduction to deployment options supported by JHipster
- Heroku
- Cloud Foundry
- Amazon Web Services
- Production deployment to Heroku cloud

Introduction to Microservice Server-Side Technologies

- Microservice applications versus monoliths
- Building blocks of a microservice architecture
- Service registry
- Service discovery
- Health check
- Dynamic routing and resiliency
- Security
- Fault tolerance and failover
- JHipster Registry
- Netflix Eureka server
- Spring cloud config server
- HashiCorp Consul
- Service discovery

- Health discovery
- K/V store
- Multiple data centers
- JHipster Gateway
- Netflix Zuul
- Hystrix
- JHipster Console
- Elasticsearch
- Logstash
- Kibana
- Zipkin
- Prometheus
- JHipster UAA server

Building Microservices with JHipster

- Application architecture
- Gateway application generation
- Converting a monolithic application to a microservice gateway
- Application generation
- Generating a new Gateway
- Gateway configuration
- JWT authentication
- How JWT works
- Microservice application – Invoice Service with MySQL database
- Application generation
- Microservice configuration
- Microservice application – notification service with NoSQL
- database
- Application generation
- Microservice configuration

Working with Microservices

- Setting up JHipster Registry locally

- Using a pre-packaged WAR file
- Building from source
- Docker mode
- Running a generated application locally
- Gateway application pages
- JHipster Registry pages
- System status
- Below renew threshold
- Instances registered
- General info and health
- Application listing page
- Metrics page
- Health page
- Configuration page
- Logs page
- Swagger API endpoints
- Running invoice and notification applications locally
- Modeling entities in JDL
- Entity generation on microservices
- Explaining the generated code
- Gateway application
- Explaining the generated pages

Deploying with Docker Compose

- Introducing microservice deployment options
- A short introduction to Docker Compose
- Kickstarting Kubernetes
- Introducing OpenShift
- Explaining Rancher
- Generated Docker Compose files
- Walking through the generated files
- Building and deploying everything to Docker locally
- Generating docker-compose files for microservices
- Features of the deployed application
- JHipster console demo
- Scaling up with Docker Swarm

Deploying to the Cloud with Kubernetes

- Generating Kubernetes configuration files with JHipster
- Walking through the generated files
- Deploying the application to Google Cloud with Kubernetes

Using React for the Client-Side

- Generating an application with React client side
- Technical stack and source code
- Technical stacks
- Using TypeScript
- State management with Redux and friends
- Routing with React Router
- HTTP requests using Axios
- Bootstrap components using Reactstrap
- Unit testing setup
- Generating source code
- Generating an entity with React client side

Best Practices with JHipster

- The next steps to pursue
- Adding a shopping cart for the application
- Improving end-to-end tests
- Improving the CI/CD pipeline
- Building a JHipster module
- Best practices to keep in mind
- Choosing a client-side framework
- Choosing a database option
- Architecture considerations
- Security considerations
- Deployment and maintenance
- General best practices

Duration and pricing

In Price Group A

Certificate

Read about our certificates

Bookings

You can download the course registration form on our home page or by clicking here

Brochure
You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions
Please email us

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.

Spring 5 Microservices

This course will help you implement the Microservices architecture in Spring Framework, Spring Boot, and Spring Cloud. Using the latest specifications of Spring that focuses on Reactive Programming, you'll be able to build modern, internet-scale Java applications in no time. The course starts off with guidelines to implement responsive microservices at scale. You will understand how Spring Boot is used to deploy server-less autonomous services by removing the need to have a heavyweight application server.

You will also learn how to go further by deploying your

microservices to Docker and managing them with Mesos. By the end of the book, you will have gained more clarity on the implementation of microservices using Spring Framework and will be able to use them in internet-scale deployments through real-world examples.

Intended Audience;

You should be familiar with Spring Core

Day 1
Demystifying Microservices

- Evolution of microservices
- What are Microservices?
- Microservices – The honeycomb analogy
- Principles of microservices
- Characteristics of microservices
- Microservices examples
- Microservices benefits
- Summary

Related Architecture Styles and Use Cases

- Service-Oriented Architecture (SOA)
- Twelve-Factor Apps
- Serverless computing
- Lambda architecture
- DevOps, Cloud, and Containers
- Reactive microservices
- Microservice use cases
- Microservices early adopters – Is there a common theme?
- Microservice frameworks

Day 2

Building Microservices with Spring Boot

- Setting up a development environment
- Spring Boot for building RESTful microservices

- Getting started with Spring Boot
- Developing a Spring Boot microservice
- Developing our first Spring Boot microservice
- HATEOAS-enabled Spring Boot microservice
- Reactive Spring Boot microservices
- Implementing security
- Enabling cross origin for microservices interactions
- Spring Boot actuators for microservices instrumentation
- Documenting microservices
- Putting it all together – Developing a customer registration microservice example

Applying Microservices Concepts

- Microservice design guidelines

Microservices Capability Model

- Microservices capability model
- Core capabilities
- Infrastructure capabilities
- Supporting capabilities
- Process and governance capabilities
- Microservices maturity model
- Entry points for adoption

Day 3

Microservices Evolution – A Case Study

- Understanding the PSS application
- Death of the monolith
- Microservices to the rescue – a planned approach for migration
- Target implementation
- Potential next steps

Scale Microservices with Spring Cloud Components

- What is Spring Cloud?

- Spring Cloud releases
- Setting up the environment for the BrownField PSS
- Spring Cloud Config
- Eureka for registration and discovery
- Zuul proxy as the API Gateway
- Streams for reactive microservices
- Protecting microservices with Spring Cloud Security
- Summarising the BrownField PSS architecture

Day 4

Logging and Monitoring Microservices

- Understanding log management challenges
- Centralized logging solution
- Selection of logging solutions
- Monitoring microservices
- Data analysis using Data Lake

Containerizing Microservices with Docker

- Understanding gaps in the BrownField PSS microservices
- What are containers?
- Difference between VM and containers
- Benefits of containers
- Microservices and containers
- Introduction to Docker
- Deploying microservices into Docker
- Running RabbitMQ on Docker
- Using the Docker registry
- Microservices on Cloud
- Running BrownField services on EC2
- Future of containerization

Day 5

Scaling Dockerized Microservices with Mesos and Marathon

- Scaling microservices
- Container orchestration

- Container orchestration with Mesos and Marathon
- Implementing Mesos and Marathon with DCOS
- Implementing Mesos and Marathon for BrownField microservices
- Preparing BrownField PSS services

Microservice Development Life Cycle

- Practice points for microservice development
- Automating development cycle
- Summary

Duration and pricing

Pricing Group A

Certificate

- Upon completion of this course we will issue you with attendance certificate to certify your attendance.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.
- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings
You can download the course registration form on our home page or by clicking here

Brochure
You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions
Please email us

Schedule
On the calendar below. If your browser doesn't display the calendar below, please click on this link or try using Google

Chrome, alternatively please enquire via our Contact Us page

Spring Core 5

This Spring Core 5 training course will introduce you to the Spring Framework and proceed to a level that you are able to prepare for the Spring Core 5 Certification Exam. We cover the latest Spring best practices, including Spring Boot for application setup and configuration.

Prerequisites

Java Advanced (SE 8 Programming / OCP)

Further training

We have a Spring 5 Core Certification Mock Exam Workshop to help you get exam-ready for the real exam

Contents

Fundamentals

- Getting Started With Spring
- Developing Web Applications
- Working With data
- Securing Spring
- Working With Configuration Properties

Spring Integration

- Providing REST Services
- Consuming REST Services
- Asynchronous Messaging
- Integrating Spring

Reactive Spring

- Introducing Reactor
- Developing Reactive Apps
- Reactive Data Persistence

Cloud Native Spring

- Discovering Services
- Managing Configuration
- Handling Failure and Latency

Deploying Spring

- Spring Boot Actuator
- Spring Administration
- Monitoring with JMX
- Spring Deployment

Duration and pricing

In pricing Group A

Certificate

Read about our certificates

Bookings

You can download the course registration form on our home page or by clicking here

Brochure
You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions
Please email us

Schedule

On the calendar below. If your browser doesn’t display the

calendar below, please click on this link or try using Google Chrome, alternatively please enquire via our Contact Us page.