O&B ACADEMY

COURSE SYLLABUS

JAVA-101

Java Fundamentals

JAVA-101

Java Fundamentals

OVERVIEW

Skill Level	:	Beginner
Suitable for	:	Anyone with basic coding experience, wanting to consolidate knowledge of Java 17
Duration	:	7 Days

This course is an introduction to Java 17 and the most commonly used libraries of the Java Standard Edition (Java SE). It can be initial preparation for the Oracle Java certification exam as it covers most of the exam coverage.

It is a pragmatic course in that it goes beyond syntax and usage but discusses basic best practices and common pitfalls. It takes a learning-by-doing approach, where trainees are taken through abundant guided and unguided hands-on exercises, with succeeding instructor analysis, code-review, and coaching.

The course culminates in a final machine problem and code review, where students combine most of what they've learned into a working program, and present to class for analysis and critique by both the instructor and the rest of the class.

LEARNING OUTCOMES

- Develop Java applications using modern IDEs.
- Write well-structured code using Object Oriented Principles (OOP).
- Use JDBC to perform SQL database queries, particularly in PostgreSQL.
- Use proper exception-handling techniques.
- Write thread-safe parallel processing code, and avoid common concurrency pitfalls.

COURSE OUTLINE

Introduction to Java

- Java as a language
- Java as a development environment
- Java as a runtime environment
- Built-in Command Line tools (CLI)
- Java Security Manager
- Java Compiler
- Javadoc
- Classpath
- Packages

Java Programming Fundamentals

- Basic Syntax
- Literals
- Data Types
- Operators
- Blocks
- Variable Scope
- Naming Conventions
- Primitives
- Casting
- Method Overloading

Control Structures

- If
- If-else
- Else If
- Switch
- Loops
- Labels
- Break
- Continue
- Data Types
- Operators
- Blocks
- Variable Scope
- Naming Conventions
- Primitives
- Casting
- Method Overloading

Arrays

- Working with Arrays in Java

Working with Objects & Classes

- Class-based Object Oriented Programming (OOP)
- Equality
- Instance Variables
- Class Variables
- Object Instantiation
- Garbage Collection
- Parameter Passing
- Statics
- Object Casting
- Comparison
- Nulls

Commonly-Used Classes

- Object
- String
- StringBuilder
- Primitive Wrappers
- BigDecimal
- String Memory Management
- Primitive Wrapper Caching
- Primitive Wrapper Performance Concerns
- Helper Classes
- Date & Time
- Legacy Date & Time

Creating Your Own Classes

- Class Declaration
- Default Constructor
- Declaring Fields
- Declaring Constructors
- final
- this
- Access Modifiers
- Referencing Other Classes
- Enums
- Getters & Setters ("JavaBean" Convention)
- Nested Classes

Inheritance & Polymorphism in Java

- Inheritance
- Polymorphism
- Method Overriding
- super
- How Polymorphism Reduces Complexity
- Annotations
- toString()
- equals()
- hashCode()
- Design By Contract
- Abstract Classes
- Abstract Methods
- Reducing Code Duplication Via Inheritance And Polymorphism
- "final" Keyword Applied To Classes And Methods
- Interfaces
- Static Methods.
- Real-life situations and recommended usage discussed for all the above.

Collections

- Lists
- Sets
- Maps
- Queues
- How to choose the correct implementation
- Proper iteration
- Performance considerations
- Copy constructors
- Helper classes
- Generics
- Real-life Development Scenarios

Anonymous Classes, Lambdas & Streams

- Anonymous classes

- Converting a one-method anonymous class into a lambda
- Using the functional interfaces (java.util.function)
- Using streams with collections

Software and Hardware Requirements

- JAVA

Exceptions

- Anonymous classes
- Converting a one-method anonymous class into a lambda
- Using the functional interfaces (java.util.function)
- Using streams with collections

Input & Output

- Byte I/O
- Character I/O
- Buffering
- Try-with-resources or closing in the finally block
- Commonly-used implementations
- Files
- Non-blocking I/O
- Channels & Buffers
- Selectors

Introduction to Threads

- Common terminologies
- Thread types
- Thread concurrency, thread scheduling
- How to create runnable classes
- How to create a thread
- Polling

Creating Threads with the Concurrency API

- Introduction to Concurrency API
- Single-thread Executor
  - using newSingleThreadExecutor, shutting down, submitting tasks
  - java.util.comcurrent.Future (CompletableFuture)
  - java.util.concurrent.Callable Functional Interface
- Concurrency with Pools

Writing Thread-Safe Code

- Understanding Thread-Safety
- Protecting Data with Atomic Classes
- Synchronized Blocks
- Synchronizing on Methods
- Lock Framework
- Orchestrating Tasks with a CyclicBarrier

Identifying Threading Problems

- Deadlock
- Starvation
- Livelock
- Race Conditions

JDBC

- Introduction to JDBC
- Acquiring connections
- Passing SQL statements to the DB
- Retrieving data

- Passing parameters to SQL
  - Batch processing
  - Retrieving Auto-Generated Keys from the DB
- Calling Stored Procedures from JDBC
- Transactions

MACHINE PROBLEM & CODE REVIEW

Towards the end of the course, participants are presented with a machine problem which challenges them to write working code that solves a specific problem, using the concepts taught in the course. The trainer will review some participants' code live during class, to illustrate further key concepts.

O&B ACADEMY

Engineering for the Real World

Enquiries

+63 2 5322 2307

training-sales@orangeandbronze.com