JAVA-SE21: Java SE 21 Programming

Course Overview

The official Oracle Java SE 21: Programming Complete course outline is strategically designed to prepare developers for the Java SE 21 Developer Professional (1Z0-830) certification. The curriculum spans fundamental syntax, core object-oriented principles, and modern Java 21 features like virtual threads and pattern matching.

Course Description

The Java SE 21: Programming Complete course is an intensive 40-hour deep dive designed to equip developers with the skills to build secure, high-performance, and maintainable enterprise applications. Centered around the latest Long-Term Support (LTS) release, the curriculum bridges foundational object-oriented logic with cutting-edge functional paradigms. Learners engage in extensive hands-on exercises to engineer scalable applications from scratch using Java's most critical APIs. Additionally, this course serves as the definitive training blueprint for passing the Oracle Certified Professional: Java SE 21 Developer (1Z0-830) certification exam

Learning Curriculum

Module 1

Java Fundamentals & Environment Setup

Ecosystem overview: Understanding the JVM, JDK, and JRE architecture.Environment configurations: Setting up the environment variables and compilation tools.Basic syntax: Working with primitives, data types, and local variable type inference (var).Operators: Evaluating arithmetic, relational, and boolean precedence rules

Module 2

Control Flow & Text Manipulation

Decision making: Writing standard if/else and advanced switch statements.Loop constructs: Utilizing while, do-while, and standard/enhanced for loops.String processing: Manipulating text blocks, StringBuilder, and the unified repeat() method

Module 3

Object-Oriented Class Design

Objects: Instantiating fields, constructors, and managing object lifecycles.Memory mechanics: Analyzing parameter passing, stack vs. heap allocation, and garbage collection.Encapsulation: Enforcing access modifiers, overloading methods, and designing immutable objects

Module 4

Advanced Inheritance & Language Types

Polymorphism: Extending classes, method overriding, and dynamic type casting.Type variants: Implementing interfaces, functional interfaces, and custom enum types.Modern typing: Structuring immutable data with record classes and configuring restrictive sealed hierarchies

Module 5

Data Structures & Functional Programming

Arrays: Initializing and managing single- and multi-dimensional matrices.Collections framework: Leveraging lists, sets, maps, and new Sequenced Collections.Lambda expressions: Writing concise functional code and using built-in predicates.Stream API: Processing pipeline data using operations like filters, mapping, reductions, and parallel sorting

Module 6

Exception Handling & Security

Error mechanics: Managing checked vs. unchecked exceptions.Resource safety: Implementing try-catch-finally structures and automated try-with-resources.Injection defense: Writing clean code to block SQL and scripting vulnerabilities

Module 7

Modern Concurrency & System Integration

Platform threads: Deploying classic multi-threaded synchronization and the ExecutorService framework.Virtual threads: Implementing massive concurrency for scalable, I/O-bound workloads.File I/O: Interfacing with files, directories, and file systems using JDK NIO.2 specifications.Database connectivity: Executing basic CRUD transactions via JDBC drivers.Modular programming: Building, packaging, and deploying applications through the Java Platform Module System (JPMS)

© 2026 Proasia Upskilling Sdn Bhd | www.proasia.my