Java Foundations and Java SE 21 Programming Complete (Lessons 1 - 6) Ed 1

JAVA

DURATION	MODULES	COURSE CODE
4 Days	**7 Lectures**	**—**

Course Overview

It provides an in-depth coverage for all core Java topics and most frequently used APIs. It also offers a set of practical exercises teaching how to build a fully functioning Java application from scratch.

What You Will Learn

Course Introduction

- Course Goals
- Audience
- Course Structure

Introduction to Java

- Objectives
- What Is Java?
- How Java Works
- Object-Oriented Principles
- Classes and Objects
- Classes
- Objects
- Inheritance
- Java APIs
- Java Keywords, Reserved Words, and Special Identifiers
- Java Naming Conventions
- Java Basic Syntax Rules
- Defining a Java Class
- Accessing Classes Across Packages
- Implementing Encapsulation with Access Modifiers

- Creating a Main Application Class
- Compiling a Java Program
- Executing a Java Program
- Comments and Documentation
- Code Snippets in Javadoc
- External Snippets
- Summary
- Practices for Lesson 1: Overview

Primitive Types, Operators, and Flow Control Statements

- Objectives
- Java Primitives
- Declaring and Initializing Primitive Variables
- Primitive Declarations and Initializations: Restrictions
- Java Operators
- Assignment and Arithmetic Operators
- Arithmetic Operations and Type Casting
- More Mathematical Operations
- Binary Number Representation
- Bitwise Logical Operators
- Equality, Relational, and Conditional Operators
- Short-Circuit Evaluation
- Flow Control Using if/else Construct
- Ternary Operator
- Flow Control Using switch Construct
- Switch -> No Fall-Through Syntax
- Switch Expressions yield a Value
- Switch Statements and Expressions Summary
- Using JShell (REPL Tool)
- JShell Commands
- Code Snippets
- Summary
- Practices for Lesson 2: Overview

Text, Date, Time, and Numeric Objects

- Objectives
- String Initialization
- String Operations
- String Indexing
- Mutable Text Objects
- Text Blocks
- Spaces, Lines, and Quotes
- Wrapper Classes for Primitives
- Representing Numbers Using BigDecimal Class
- Method Chaining

- Local Date and Time API
- More Local Date and Time Operations
- Instants, Durations, and Periods
- Zoned Date and Time
- Representing Languages and Countries
- Formatting and Parsing Numeric Values
- Formatting and Parsing Date and Time Values
- Localizable Resources
- Formatting Message Patterns
- Formatting and Localization: Summary
- Summary
- Practices for Lesson 3: Overview

Classes and Objects

- Objectives
- UML: Introduction
- Modeling Classes
- Modeling Interactions and Activities
- Designing Classes
- Creating Objects
- Defining Instance Variables
- Defining Instance Methods
- Object Creation and Access: Example
- Local Variables and Recursive Object Reference
- Local Variable Type Inference
- Defining Constants
- Static Context
- Accessing Static Context
- Combining Static and Final
- Other Static Context Use Cases
- IntelliJ IDE: Introduction
- Summary
- Practices for Lesson 4: Overview

Improved Class Design

- Objectives
- Overload Methods
- Variable Number of Arguments
- Defining Constructors
- Reusing Constructors
- Access Modifiers: Summary
- Defining Encapsulation
- Defining Immutability
- Constants and Immutability
- Enumerations

- Complex Enumerations
- Java Memory Allocation
- Parameter Passing
- Java Memory Cleanup
- Summary
- Practices for Lesson 5: Overview

Implement Inheritance and Use Records

- Objectives
- Extending Classes
- Object Class
- Reusing Parent Class Code Through Inheritance
- Instantiating Classes and Accessing Objects
- Rules of Reference Type Casting
- Verifying Object Type Before Casting the Reference
- Pattern Matching for instanceof
- Reference Code Within the Current or Parent Object
- Defining Subclass Constructors
- Class and Object Initialization: Summary
- Overriding Methods and Using Polymorphism
- Reusing Parent Class Logic in Overwritten Method
- Defining Abstract Classes and Methods
- Defining Final Classes and Methods
- Sealed Classes and Interfaces
- Overriding Object Class Operations: toString
- Overriding Object Class Operations: equals
- Override Object Class Operations: hashCode
- Comparing String Objects
- Java Records
- Custom Record Constructors
- Record Patterns
- Pattern Matching for switch
- Factory Methods
- Summary
- Practices for Lesson 6: Overview