

Java Programming – Core to Advanced

BY S. PRADEEP KUMAR

Duration: 20 Hours Intensive Training

Format: 100% Practical & Hands-on

Level: Core to Advanced Enterprise

WHY THIS COURSE MATTERS

Mastering Java requires more than understanding syntax—it demands deep algorithmic clarity, robust enterprise design patterns, and hands-on exposure to modern frameworks. This 20-hour intensive, practical curriculum is engineered to transform learners into job-ready backend engineers. From core memory mechanics and modern functional paradigms (Java 8+) to concurrency, JDBC, and test-driven development (JUnit 5 & Mockito), every module equips you with industry-grade software engineering skills.

Chapter 1: Java Basics & Foundational Programming

Java Setup & Execution

- JDK, JRE, and JVM Architecture
- Bytecode Execution & JIT Compiler
- Main Method Structure & Execution Flow

Variables & Operators

- Data Types & Variable Scopes
- Implicit & Explicit Type Casting
- Arithmetic, Logical & Ternary Operators

Control Flow

- Decision Making (if-else, switch expressions)
- Loops (for, while, do-while)
- Jump Statements (break, continue, return)

Arrays & Memory

- 1D & Multi-Dimensional Arrays
- Array Traversal & java.util.Arrays Utility
- Stack vs. Heap Memory Allocation

Chapter 2: Strings & Utility Classes

String Fundamentals

- String Literals vs. new Keyword
- String Constant Pool (SCP) Mechanics
- String Immutability & Performance

String Operations

- Core Methods (substring, split, replace, equals)
- StringBuilder vs. StringBuffer Comparison

Wrapper Classes

- Primitive Wrappers (Integer, Double, etc.)
- Autoboxing & Unboxing Mechanics

Chapter 3: Object-Oriented Programming (OOPs)

Class & Object

- Defining Fields & Instance Methods
- Object Instantiation & Lifecycle
- Instance vs. Static Members

Constructor & 'this' Keyword

- Default & Parameterized Constructors
- Constructor Chaining
- 'this' Reference Usage & Context

Encapsulation & Access Control

- Access Modifiers (private, public, protected, default)
- Getters, Setters & Data Hiding
- Packages & Import Management

Inheritance & 'super' Keyword

- Extending Classes
- Method Overriding
- 'super' Keyword Usage

Polymorphism & 'final' Keyword

- Method Overloading vs. Overriding
- Dynamic Method Dispatch
- 'final' Variables, Methods & Classes

Abstraction

- Abstract Classes & Abstract Methods
- Interfaces & Concrete Implementations
- Default & Static Interface Methods

Chapter 4: Collections Framework & Generics

Generics

- Generic Classes & Generic Methods

List Interface

- ArrayList
- LinkedList
- Stack
- Vector

Set Interface

- HashSet & LinkedHashSet
- TreeSet & Custom Sorting

Map Interface

- HashMap & Hash Collisions
- hashCode() and equals() Contract
- TreeMap

Iteration & Sorting

- Iterator & ListIterator
- Comparable vs. Comparator

Chapter 5: Exception Handling & File I/O

Exception Hierarchy

- Checked vs. Unchecked Exceptions
- Throwable, Exception, and Error

Handling Exceptions

- try-catch-finally Blocks
- Multi-catch & Nested try
- Custom Exceptions (throw, throws)

Resource Management & Files

- try-with-resources & AutoCloseable
- BufferedReader & BufferedWriter

Chapter 6: Modern Java (Java 8+)

Lambda Expressions

- Lambda Syntax & Behavior Passing
- @FunctionalInterface Annotation
- Built-in Interfaces (Predicate, Function, Consumer, Supplier)
- Method References (::)

Null Safety

- Optional<T> Usage
- Avoiding NullPointerException

Stream API

- Stream Creation
- Intermediate Operations (filter, map, sorted)
- Terminal Operations (collect, reduce, forEach)

Chapter 7: Multithreading, Concurrency & Clean Code

Multithreading Basics

- Thread Class & Runnable Interface
- Thread Lifecycle & States
- synchronized & Thread Safety

Concurrency Utilities

- ExecutorService & Thread Pools
- Callable & Future
- Atomic Variables

Clean Code & Design Principles

- Overview of SOLID Principles
- Creational Patterns (Singleton, Factory, Builder)

Chapter 8: Database (JDBC) & Testing (JUnit & Mockito)

Database & SQL Fundamentals

- Basic DDL, DML, and DQL Queries
- SQL Joins & Foreign Keys

JDBC Integration

- DriverManager & Connection
- PreparedStatement & ResultSet
- Basic Transaction Management

Unit Testing with JUnit 5

- @Test, @BeforeEach, @AfterEach Annotations
- Assertions (assertEquals, assertTrue)

Mocking with Mockito

- @Mock and @InjectMocks
- Stubbing (when().thenReturn())
- @Spy Annotation