# Mastering Java

## 1. Introduction to Java

* **1.1 History and Evolution of Java**
* **1.2 Features of Java**
  + Platform Independence
  + Object-Oriented Principles
  + Security and Robustness
  + Multithreading and Concurrency
  + High Performance and Scalability
* **1.3 The Java Virtual Machine (JVM)**
  + Bytecode Execution
  + Just-In-Time Compilation
* **1.4 Setting Up the Development Environment**
  + Installing Java Development Kit (JDK)
  + Configuring Environment Variables
  + Introduction to Integrated Development Environments (IDEs)
* **1.5 Writing Your First Java Program**
  + Hello World Example
  + Compiling and Running Java Programs
* **1.6 Understanding Java Program Structure**

## 2. Java Basics

* **2.1 Syntax and Basic Constructs**
* **2.2 Data Types and Variables**
  + Primitive Data Types
  + Reference Data Types
  + Variable Declaration and Initialization
  + Type Casting and Conversion
* **2.3 Operators**
  + Arithmetic Operators
  + Assignment Operators
  + Comparison Operators
  + Logical Operators
  + Bitwise Operators
* **2.4 Control Flow Statements**
  + Conditional Statements (if, else, switch)
  + Looping Constructs (for, while, do-while)
  + Break and Continue Statements
* **2.5 Arrays**
  + One-Dimensional Arrays
  + Multidimensional Arrays
  + Array Initialization and Manipulation
* **2.6 Strings and String Handling**
  + String Class and Methods
  + StringBuilder and StringBuffer
* **2.7 Input and Output Basics**
  + Scanner Class
  + Console Input and Output

## 3. Object-Oriented Programming in Java

* **3.1 Classes and Objects**
  + Defining Classes
  + Creating Objects
  + Class Members (Fields and Methods)
* **3.2 Methods and Constructors**
  + Method Overloading
  + Constructors and Constructor Overloading
  + The this Keyword
* **3.3 Inheritance**
  + Extending Classes
  + The super Keyword
  + Method Overriding
* **3.4 Polymorphism**
  + Compile-Time Polymorphism
  + Runtime Polymorphism
* **3.5 Encapsulation**
  + Access Modifiers (public, private, protected)
  + Getters and Setters
* **3.6 Abstraction**
  + Abstract Classes
  + Interfaces
  + Difference Between Abstract Classes and Interfaces
* **3.7 Packages and Access Control**
  + Organizing Code with Packages
  + Import Statements

## 4. Advanced Object-Oriented Concepts

* **4.1 Nested and Inner Classes**
  + Static Nested Classes
  + Inner Classes
  + Anonymous Inner Classes
* **4.2 Enumerations (Enums)**
  + Defining Enums
  + Enum Methods and Usage
* **4.3 Generics**
  + Generic Classes and Methods
  + Type Parameters and Bounds
  + Wildcards in Generics
* **4.4 Annotations**
  + Built-in Annotations (@Override, @Deprecated)
  + Custom Annotations
  + Meta-Annotations
* **4.5 Records (Java 16+)**
  + Creating Immutable Data Classes
  + Record Components and Methods
* **4.6 Sealed Classes (Java 17+)**
  + Restricting Class Hierarchies
  + Permitted Subclasses

## 5. Exception Handling

* **5.1 Understanding Exceptions**
  + Checked vs. Unchecked Exceptions
* **5.2 Try-Catch-Finally Blocks**
* **5.3 Throwing Exceptions**
  + The throw and throws Keywords
* **5.4 Custom Exceptions**
  + Creating User-Defined Exception Classes
* **5.5 Best Practices in Exception Handling**

## 6. Java Collections Framework

* **6.1 Introduction to Collections**
  + Collection Interfaces and Classes
* **6.2 List Interface**
  + ArrayList
  + LinkedList
* **6.3 Set Interface**
  + HashSet
  + LinkedHashSet
  + TreeSet
* **6.4 Map Interface**
  + HashMap
  + LinkedHashMap
  + TreeMap
* **6.5 Queue and Deque Interfaces**
  + PriorityQueue
  + ArrayDeque
* **6.6 Iterators and Enhanced for-loop**
* **6.7 Comparator and Comparable Interfaces**
* **6.8 Collections Utility Class**
  + Sorting and Searching
  + Thread-Safe Collections

## 7. Java Input/Output (I/O)

* **7.1 Streams in Java**
  + Byte Streams
  + Character Streams
* **7.2 File Handling**
  + Reading and Writing Files
  + Buffered Streams
* **7.3 Serialization**
  + Serializable Interface
  + ObjectInputStream and ObjectOutputStream
* **7.4 New I/O (NIO)**
  + Buffers and Channels
  + FileChannel and ByteBuffer
* **7.5 NIO.2 Enhancements (Java 7+)**
  + Path and Files Classes
  + Asynchronous File I/O

## 8. Multithreading and Concurrency

* **8.1 Introduction to Multithreading**
  + Thread Class and Runnable Interface
* **8.2 Thread Lifecycle and States**
* **8.3 Synchronization**
  + Synchronized Methods and Blocks
  + Intrinsic Locks and Reentrant Locks
* **8.4 Inter-Thread Communication**
  + wait(), notify(), and notifyAll()
* **8.5 Concurrency Utilities**
  + Executors and Thread Pools
  + Callable and Future Interfaces
  + Concurrent Collections
* **8.6 Atomic Variables and Locks**
* **8.7 Fork/Join Framework**
* **8.8 CompletableFuture and Asynchronous Programming**
* **8.9 Virtual Threads (Java 21)**
  + Project Loom Overview
  + Creating and Managing Virtual Threads
* **8.10 Structured Concurrency (Java 21)**
  + Managing Task Lifecycles
  + Error Handling in Concurrent Tasks

## 9. Java Memory Management

* **9.1 JVM Memory Model**
  + Heap and Stack Memory
  + Method Area and Runtime Constant Pool
* **9.2 Garbage Collection**
  + Garbage Collection Algorithms
  + Tuning Garbage Collection
* **9.3 Memory Leaks and Optimization**
  + Common Causes
  + Detection Tools

## 10. Functional Programming in Java

* **10.1 Lambda Expressions**
  + Syntax and Functional Interfaces
* **10.2 Method References**
  + Types and Usage
* **10.3 Stream API**
  + Creating Streams
  + Intermediate Operations (filter, map)
  + Terminal Operations (collect, reduce)
* **10.4 Optional Class**
  + Avoiding NullPointerExceptions
* **10.5 Date and Time API (Java 8+)**
  + LocalDate, LocalTime, LocalDateTime
  + DateTimeFormatter

## 11. New Features from Java 9 to Java 21

* **11.1 Java 9 Enhancements**
  + Module System (Project Jigsaw)
  + JShell (Read-Eval-Print Loop)
  + Private Methods in Interfaces
* **11.2 Java 10 Updates**
  + Local Variable Type Inference (var)
* **11.3 Java 11 Additions**
  + New String Methods
  + HTTP Client API
  + Running Java Files without Compilation
* **11.4 Java 12-17 Features**
  + Switch Expressions
  + Text Blocks
  + Records
  + Pattern Matching for instanceof
  + Sealed Classes
* **11.5 Java 21 Innovations**
  + Virtual Threads
  + Structured Concurrency
  + Sequenced Collections
  + String Templates (Preview)
* **11.6 Deprecated and Removed Features**
* **11.7 Migration Tips**

## 12. Java Networking

* **12.1 Networking Basics**
  + Sockets and IP Addresses
* **12.2 Working with URLs and URIs**
* **12.3 Socket Programming**
  + TCP Sockets
  + UDP Sockets
* **12.4 HTTP Communication**
  + HttpURLConnection
  + New HTTP Client API (Java 11+)
* **12.5 Multicast Sockets**

## 13. Java Database Connectivity (JDBC)

* **13.1 Introduction to JDBC**
  + JDBC Drivers
* **13.2 Connecting to Databases**
* **13.3 Executing SQL Statements**
  + Statement and PreparedStatement
* **13.4 ResultSet and Metadata**
* **13.5 Transaction Management**
* **13.6 Batch Updates**
* **13.7 Connection Pooling**

## 14. Java Annotations

* **14.1 Built-in Annotations**
  + @Override, @Deprecated, @SuppressWarnings
* **14.2 Custom Annotations**
  + Defining and Using Custom Annotations
* **14.3 Annotation Processing**
  + Retention Policies
  + Reflection with Annotations

## 15. Java Reflection API

* **15.1 Introduction to Reflection**
* **15.2 The Class Class**
* **15.3 Accessing Fields, Methods, and Constructors**
* **15.4 Dynamic Method Invocation**
* **15.5 Reflection Use Cases and Best Practices**

## 16. Java Design Patterns

* **16.1 Creational Patterns**
  + Singleton
  + Factory Method
  + Builder
* **16.2 Structural Patterns**
  + Adapter
  + Decorator
  + Proxy
* **16.3 Behavioral Patterns**
  + Observer
  + Strategy
  + Command
* **16.4 Best Practices in Applying Design Patterns**

## 17. Java GUI Programming

* **17.1 Introduction to JavaFX**
* **17.2 Setting Up JavaFX Environment**
* **17.3 Scene Graph and Application Structure**
* **17.4 Controls and Layouts**
* **17.5 Event Handling**
* **17.6 Styling with CSS**
* **17.7 FXML and Scene Builder**

## 18. Best Practices and Code Optimization

* **18.1 Writing Clean and Maintainable Code**
* **18.2 Refactoring Techniques**
* **18.3 Performance Tuning**
  + Profiling Java Applications
  + Memory Management
* **18.4 Effective Use of Data Structures**
* **18.5 Code Conventions and Style Guides**
* **18.6 Documentation with Javadoc**

## 19. Testing in Java

* **19.1 Introduction to Unit Testing**
* **19.2 JUnit Framework**
  + Writing Test Cases
  + Assertions and Test Fixtures
* **19.3 Mocking with Mockito**
  + Creating Mocks
  + Verifying Behavior
* **19.4 Test-Driven Development (TDD)**
* **19.5 Continuous Integration and Deployment**

## 20. Build Tools and Dependency Management

* **20.1 Introduction to Build Automation**
* **20.2 Apache Maven**
  + Project Object Model (POM)
  + Dependencies and Repositories
  + Build Lifecycle and Plugins
* **20.3 Gradle Build Tool**
  + Build Scripts
  + Dependency Management
* **20.4 Comparing Maven and Gradle**

## 21. Deployment and Packaging

* **21.1 Packaging Java Applications**
  + JAR Files
  + Executable JARs
* **21.2 Native Packaging**
  + Using jpackage Tool
* **21.3 Java Modules and Deployment**

## 22. Java and the Web

* **22.1 Java Servlets**
  + Servlet Lifecycle
  + Handling Requests and Responses
* **22.2 JavaServer Pages (JSP)**
  + JSP Syntax
  + Expression Language
* **22.3 RESTful Web Services with JAX-RS**
  + Annotations and Resource Classes
* **22.4 Introduction to Spring Framework**
  + Dependency Injection
  + Spring MVC and REST
* **22.5 Building Microservices with Spring Boot**

## 23. Security in Java

* **23.1 Java Security Overview**
* **23.2 Cryptography**
  + Encryption and Decryption
  + Message Digests and Signatures
* **23.3 Secure Communication**
  + SSL/TLS and HTTPS
* **23.4 Authentication and Authorization**
* **23.5 Secure Coding Practices**
  + Input Validation
  + Access Control
* **23.6 Java Security Manager**

## 24. Advanced Topics

* **24.1 JVM Internals and Bytecode**
* **24.2 Custom Class Loaders**
* **24.3 Dynamic Proxies**
  + Creating Proxies at Runtime
* **24.4 Java Native Interface (JNI)**
  + Interfacing with Native Code
* **24.5 Scripting with Java**
  + Using Nashorn JavaScript Engine
  + Integrating Other Languages
* **24.6 Performance Tuning Tools**
  + VisualVM
  + Java Flight Recorder

## 25. Future of Java

* **25.1 OpenJDK and Community Process**
* **25.2 Upcoming Features and Roadmap**
* **25.3 Contributing to Open Source**
* **25.4 Trends in Java Development**

## 26. Appendices

* **A. Glossary of Terms**
* **B. Resources for Further Learning**
  + Official Documentation
  + Books and Tutorials
  + Online Communities
* **C. Setting Up Development Environments**
  + Eclipse
  + IntelliJ IDEA
  + Visual Studio Code
* **D. Java Certifications**
  + Certification Paths
  + Preparation Tips

————————

This comprehensive table of contents is designed to guide you from a beginner to an expert in Java programming, covering all essential topics and the latest features up to Java 21.

#software/languages/java