Here's a **day-wise detailed structure** for learning **Java from Basics to Advanced** in a structured manner. This plan is designed for **50 days**, ensuring steady progress with **practical implementation**.

**Day-Wise Java Learning Roadmap (50 Days)**

**Week 1: Getting Started with Java (Basics)**

**Day 1: Introduction & Setup**

* Introduction to Java and its Features
* Setting up Java Development Kit (JDK) & IDE (Eclipse/IntelliJ/VS Code)
* Understanding JVM, JRE, and JDK

**Day 2: Writing Your First Program**

* Writing Your First Java Program (Hello World)
* Understanding the main() method
* Java Syntax & Code Structure

**Day 3: Data Types & Variables**

* Primitive and Non-Primitive Data Types
* Variable Declaration, Scope & Lifetime
* Type Casting (Implicit & Explicit)
* instanceof Operator

**Day 4: Operators & Control Statements**

* Arithmetic, Logical, Bitwise, Relational Operators
* Assignment & Ternary Operators
* if-else, switch-case Statements

**Day 5: Loops & Iterations**

* for, while, and do-while Loops
* Nested Loops & Loop Control Statements (break, continue)

**Week 2: Core Java & OOP Concepts**

**Day 6: Object-Oriented Programming (OOP) - Basics**

* Introduction to OOP Principles
* Classes, Objects, and Methods

**Day 7: Constructors & Access Modifiers**

* Default, Parameterized & Copy Constructors
* this and super keywords
* Access Specifiers (public, private, protected, default)

**Day 8: Inheritance & Polymorphism**

* Types of Inheritance (Single, Multilevel, Hierarchical, Multiple via Interface)
* Method Overloading & Overriding

**Day 9: Encapsulation & Abstraction**

* Getters & Setters
* Abstract Classes vs Interfaces

**Day 10: Static Keyword & Wrapper Classes**

* Static Variables, Methods, and Blocks
* Wrapper Classes (Integer, Double, Boolean, etc.)
* Autoboxing & Unboxing

**Week 3: Exception Handling & Collections**

**Day 11: Exception Handling Basics**

* Try-Catch-Finally
* Throw & Throws

**Day 12: Custom Exceptions & Advanced Handling**

* Custom Exception Handling
* Multi-Catch Block
* Exception Propagation & Chained Exceptions

**Day 13: Introduction to Java Collections Framework (JCF)**

* List, Set, and Map Overview
* Difference Between ArrayList, LinkedList, and Vector

**Day 14: Working with Lists**

* ArrayList, LinkedList, Vector (Comparison & Use Cases)

**Day 15: Working with Sets & Maps**

* HashSet, LinkedHashSet, TreeSet
* HashMap, LinkedHashMap, TreeMap, Hashtable

**Week 4: Multithreading & File Handling**

**Day 16: Multithreading Basics**

* Thread Class & Runnable Interface
* Thread Lifecycle

**Day 17: Advanced Multithreading**

* Synchronization
* Inter-Thread Communication

**Day 18: Callable, Future & Thread Priority**

* Volatile Keyword
* Atomic Variables

**Day 19: File Handling (I/O Streams)**

* File Handling Basics
* FileReader, FileWriter, BufferedReader, BufferedWriter

**Day 20: Advanced File Handling**

* Serialization & Deserialization
* Byte Streams & Character Streams
* NIO Package (Files, Path, FileChannel)

**Week 5: JDBC & Java 8 Features**

**Day 21: Java Database Connectivity (JDBC)**

* Introduction to JDBC
* Connecting Java with MySQL/PostgreSQL

**Day 22: CRUD Operations in JDBC**

* Creating Tables & Performing CRUD Operations
* Statement vs PreparedStatement

**Day 23: Connection Pooling & Transactions**

* Connection Pooling in JDBC
* Handling Transactions

**Day 24: Java 8 Features - Lambda & Functional Interfaces**

* Lambda Expressions
* Functional Interfaces (Predicate, Consumer, Supplier, etc.)

**Day 25: Java 8 Streams API & Optional Class**

* Streams & Parallel Streams
* Collectors API (groupingBy, joining(), partitioningBy)
* Optional Class

**Week 6: Advanced Topics & Java 9+ Features**

**Day 26: Java Reflection API**

* Introduction to Reflection
* Creating Objects & Accessing Private Fields

**Day 27: Garbage Collection & Memory Management**

* JVM Memory Model
* Types of Garbage Collectors (G1, CMS, etc.)
* Finalize Method (Deprecated in Java 9)

**Day 28: Annotations & Java Modules (JPMS)**

* Custom Annotations (@Retention, @Target)
* Java Modules & module-info.java

**Day 29: Design Patterns (Part 1)**

* Singleton, Factory, Observer

**Day 30: Design Patterns (Part 2)**

* Dependency Injection & IoC

**Week 7: Java Frameworks (Spring & Hibernate)**

**Day 31: Spring Core & Spring Boot Basics**

* Introduction to Spring Framework
* Spring Boot Overview

**Day 32: Spring Boot Dependency Injection**

* Spring Boot Annotations (@Component, @Autowired, @Service)

**Day 33: Building REST APIs with Spring Boot**

* REST API Development
* Handling Requests & Responses

**Day 34: Spring Security Basics**

* Authentication & Authorization in Spring

**Day 35: Introduction to Hibernate (ORM)**

* What is ORM?
* Hibernate vs JDBC

**Week 8: Full-Stack Development with Java**

**Day 36: Spring Data JPA & Repository Pattern**

* CRUD Operations using Spring Data JPA

**Day 37: Microservices with Spring Boot**

* Introduction to Microservices
* RESTful Web Services

**Day 38: Spring Cloud (Microservices Architecture)**

* Service Discovery & API Gateway

**Day 39: Maven & Gradle**

* Build & Dependency Management

**Day 40: Integrating Java with Frontend (React/Angular)**

* REST API Integration with Frontend

**Week 9: Final Steps & Real-World Java Development**

**Day 41: Java Code Optimization & Debugging**

* Best Practices for Java Code
* Debugging Java Apps

**Day 42: Java Application Profiling**

* Tools: JVisualVM, JConsole

**Day 43: Building a Mini Project**

* Java Project using Core Java & JDBC

**Day 44: Building a REST API Project**

* End-to-End REST API Development

**Day 45: Full-Stack Project**

* Spring Boot + React/Angular

With in 30 days --

**Week 1: Java Basics & OOP**

**Day 1: Introduction & Setup**

* Java Overview, JDK, JRE, JVM
* Setting up Java (Eclipse/IntelliJ/VS Code)

**Types & Variables**

* Primitive & Non-primitive Types
* Type Casting & instanceof

**Operators & Control Statements**

* Arithmetic, Logical, Bitwise Operators
* if-else, switch-case

**Loops & Iterations**

* for, while, do-while loops
* Loop control statements (break, continue)

**Day 2: Object-Oriented Programming (OOP)**

* Classes & Objects
* Constructors (this, super)

**Day 3: Inheritance & Polymorphism**

* Method Overloading & Overriding
* Types of Inheritance

**Day 4 Advanced OOP**

**Encapsulation & Abstraction**

* Getters & Setters, Abstract Classes, Interfaces

**Day 5: Static, Wrapper Classes & Inner Classes**

* Static Methods, Variables, Blocks
* Wrapper Classes (Integer, Double, Boolean)

**Day 6: Exception Handling**

* Try-Catch, Finally
* Throw, Throws & Custom Exceptions

**Day 7: Java Collections Framework (JCF) Overview**

* List, Set, Map Differences
* ArrayList vs LinkedList

**Day 8: Lists & Sets in Detail**

* ArrayList, LinkedList, HashSet, TreeSet

**Day 9-10: Maps & Advanced Collections**

* HashMap, LinkedHashMap, TreeMap, Hashtable

**Day 11: Java Generics & Streams API**

* Generic Classes & Methods
* Introduction to Streams

**Day 12: Multithreading Basics**

* Thread Class & Runnable Interface
* Thread Lifecycle

**Day 13: Advanced Multithreading**

* Synchronization & Inter-Thread Communication

**Day 14: Java File Handling (I/O)**

* FileReader, FileWriter, BufferedReader

**Day 15: Serialization & NIO Package**

* Object Streams, FileChannel, Paths

**Day 17: Java Database Connectivity (JDBC)**

* Connecting Java with MySQL/PostgreSQL
* Statement vs PreparedStatement

**Day 18: JDBC CRUD Operations & Connection Pooling**

* Performing CRUD Operations
* Handling Transactions

**Day 19: Java 8 Features - Lambda & Functional Interfaces**

* Lambda Expressions
* Functional Interfaces (Predicate, Consumer, Supplier)

**Day 20: Streams API & Optional Class**

* Collectors API (groupingBy, joining)
* Streams & Parallel Streams

**Day 21: Java Reflection & Garbage Collection**

* Reflection API
* JVM Memory Management & Garbage Collectors

**Day 22: Spring Boot Basics**

* Introduction to Spring Boot
* REST API Development

**Day 23: Spring Data JPA & Hibernate**

* ORM vs JDBC
* CRUD Operations using Spring Data JPA

**Day 24-30: Spring Boot**

**Mini Project**

**Structured Java Learning Plan with Mini-Project**

**🔹 Week 1: Java Basics & OOP (Fundamentals & Core Java)**

**Day 1: Introduction to Java & Basic Syntax**

- Java Overview: JVM, JDK, JRE  
- Setting up Java (IntelliJ/Eclipse/VS Code)  
- Writing your first Java program (Hello World!)  
- Data Types & Variables (Primitive & Non-primitive)  
- Type Casting & instanceof

**Day 2: Operators & Control Flow**

- Arithmetic, Logical, Bitwise Operators  
- if-else, switch-case statements  
- Loops (for, while, do-while)  
- Loop control (break, continue)

**Day 3: Strings & Arrays**

- String operations (length(), substring(), replace(), split())  
- StringBuilder & StringBuffer (Mutable Strings)  
- Arrays (1D & 2D Arrays)  
- Array operations (sort(), search(), copy())

**Day 4: Object-Oriented Programming (OOP) - Part 1**

- **Classes & Objects**: Creating and using classes  
- **Constructors**: Default & Parameterized (this, super)  
- **Encapsulation**: Getters & Setters

**Day 5: Object-Oriented Programming (OOP) - Part 2**

- **Inheritance**: Single, Multilevel, Hierarchical  
- **Polymorphism**: Method Overloading & Method Overriding  
- **Abstraction**: Abstract classes & Interfaces

**🔹 Week 2: Advanced Java (Collections, Exception Handling, and File Handling)**

**Day 6: Java Collections Framework (JCF) - Part 1**

- Collections Overview: List, Set, Map  
- ArrayList vs LinkedList (Performance Comparison)  
- HashSet & TreeSet (Unique Elements & Sorting)

**Day 7: Java Collections Framework (JCF) - Part 2**

- HashMap, TreeMap, LinkedHashMap  
- Iterating over Maps (forEach(), entrySet())  
- Sorting Collections using Comparator & Comparable

**Day 8: Exception Handling & Debugging**

- try-catch-finally  
- throw & throws  
- Custom Exceptions  
- Debugging Java Code

**Day 9: File Handling & Serialization**

- Reading & Writing Files (FileReader, FileWriter)  
- BufferedReader & BufferedWriter (Efficient File Handling)  
- Object Serialization & Deserialization

**Day 10: Multithreading & Concurrency Basics**

- Creating Threads (Thread Class & Runnable Interface)  
- Thread Lifecycle & Synchronization  
- Inter-thread communication (wait(), notify())

**🔹 Week 3: Java 8+ Features & Database (JDBC & Spring Boot Basics)**

**Day 11: Java 8 Features - Lambda & Streams API**

- Lambda Expressions  
- Functional Interfaces (Predicate, Consumer, Supplier)  
- Streams API (map(), filter(), collect())

**Day 12: Java 8 Streams & Optional Class**

- Working with Streams (groupingBy(), joining())  
- Optional Class (Avoiding NullPointerException)

**Day 13: Introduction to JDBC & Database Connectivity**

- Connecting Java with MySQL  
- Executing SQL Queries in Java  
- Statement vs PreparedStatement

**Day 14: JDBC CRUD Operations & Transactions**

- Performing **CRUD (Create, Read, Update, Delete)** Operations  
- Handling Transactions (commit(), rollback())

**Day 15: Introduction to Spring Boot & REST APIs**

- Setting up a Spring Boot Project  
- Creating a Simple REST API  
- Handling HTTP Requests (GET, POST, PUT, DELETE)

**🔹 Week 4: Mini-Project Implementation & Advanced Topics**

**Mini-Project: Employee Management System**

🔹 **Backend:** Spring Boot (CRUD API, JWT Authentication, MySQL)  
🔹 **Frontend:** Basic UI (React/Angular or Java Swing for UI)  
🔹 **Features:**  
-User Authentication (Login/Register with JWT)  
-Employee CRUD Operations (Add, Edit, Delete, View)  
-Role-Based Access Control

**Day 16-17: Building the Backend (Spring Boot + Database)**

- Creating the Spring Boot Project  
- Connecting Spring Boot to MySQL  
- Creating Employee Entity & CRUD API

**Day 18: Securing APIs with JWT Authentication**

- Implementing User Authentication & Role-Based Access  
- Generating JWT Tokens for Secure Access

**Day 19: Integrating the Frontend (React/Angular/Java Swing)**

- Connecting Frontend with Backend API  
- Displaying Data in UI (Employee List, Forms)

**Day 20: Deployment, Testing & Final Review**

- Testing REST APIs with Postman  
- Deploying with Docker  
- Revising Java Concepts & Mock Interview

**🔹 Structured Java Learning Plan**

**🔹 Week 1: Java Basics & OOP**

**Day 1: Java Introduction & Setup**  
- Java Overview: JVM, JDK, JRE  
- Setting up Java in IntelliJ/Eclipse/VS Code  
- Writing a Simple Java Program (Hello World)  
- Data Types & Variables (Primitive & Non-primitive)  
- Type Casting (widening & narrowing)

**Day 2: Operators & Control Flow**  
- Arithmetic, Logical, Bitwise Operators  
- if-else, switch-case statements  
- Loops (for, while, do-while)  
- Loop control statements (break, continue)

**Day 3: Strings & Arrays**  
- String operations (length(), substring(), replace(), split())  
- StringBuilder & StringBuffer (Mutable Strings)  
- Arrays (1D & 2D Arrays)  
- Array operations (sort(), search(), copy())

**Day 4: Object-Oriented Programming (OOP) - Part 1**  
- **Classes & Objects**: Creating and using classes  
- **Constructors**: Default & Parameterized (this, super)  
- **Encapsulation**: Getters & Setters

**Day 5: Object-Oriented Programming (OOP) - Part 2**  
- **Inheritance**: Single, Multilevel, Hierarchical  
- **Polymorphism**: Method Overloading & Method Overriding  
- **Abstraction**: Abstract classes & Interfaces

**🔹 Week 2: Advanced Java (Collections, Exception Handling, and File Handling)**

**Day 6: Java Collections Framework (JCF) - Part 1**  
- Collections Overview: List, Set, Map  
- ArrayList vs LinkedList (Performance Comparison)  
- HashSet & TreeSet (Unique Elements & Sorting)

**Day 7: Java Collections Framework (JCF) - Part 2**  
- HashMap, TreeMap, LinkedHashMap  
- Iterating over Maps (forEach(), entrySet())  
- Sorting Collections using Comparator & Comparable

**Day 8: Exception Handling & Debugging**  
- try-catch-finally  
- throw & throws  
- Custom Exceptions  
- Debugging Java Code

**Day 9: File Handling & Serialization**  
- Reading & Writing Files (FileReader, FileWriter)  
- BufferedReader & BufferedWriter (Efficient File Handling)  
- Object Serialization & Deserialization

**Day 10: Multithreading & Concurrency Basics**  
- Creating Threads (Thread Class & Runnable Interface)  
- Thread Lifecycle & Synchronization  
- Inter-thread communication (wait(), notify())

**🔹 Week 3: Java 8+ Features & Database (JDBC & Spring Boot Basics)**

**Day 11: Java 8 Features - Lambda & Streams API**  
- Lambda Expressions  
- Functional Interfaces (Predicate, Consumer, Supplier)  
- Streams API (map(), filter(), collect())

**Day 12: Java 8 Streams & Optional Class**  
- Working with Streams (groupingBy(), joining())  
- Optional Class (Avoiding NullPointerException)

**Day 13: Introduction to JDBC & Database Connectivity**  
- Connecting Java with MySQL  
- Executing SQL Queries in Java  
- Statement vs PreparedStatement

**Day 14: JDBC CRUD Operations & Transactions**  
- Performing **CRUD (Create, Read, Update, Delete)** Operations  
- Handling Transactions (commit(), rollback())

**Day 15: Introduction to Spring Boot & REST APIs**  
- Setting up a Spring Boot Project  
- Creating a Simple REST API  
- Handling HTTP Requests (GET, POST, PUT, DELETE)

**🔹 Week 4: Advanced Topics & Practical Application**

**Day 16: Java Reflection & Memory Management**  
- Reflection API (Accessing Classes & Methods Dynamically)  
- JVM Memory Model (Heap, Stack)  
- Garbage Collection (GC Algorithms & Optimization)

**Day 17: Functional Programming & Streams in Depth**  
- Functional Programming Concepts  
- Streams API Advanced Operations (flatMap(), reduce())

**Day 18: Working with Microservices & Spring Cloud**  
- Introduction to Microservices  
- Service Discovery & API Gateway  
- REST API Communication (Feign Client, RestTemplate)

**Day 19: Unit Testing & Performance Optimization**  
- Writing JUnit Test Cases  
- Debugging & Performance Optimization Techniques  
- Profiling Java Applications