

Part 1



JAVA BACKEND COMPLETE ROADMAP

>>>

Core Java Fundamentals

- JVM Architecture (Classloader, Runtime Memory Areas, Execution Engine)
- JDK vs JRE vs JVM
- Data Types & Variables
- Operators
- Control Flow Statements (if, switch, loops)
- Type Casting (Primitive & Object)
- Strings & String Pool
- Immutable Objects

>>>

OOP (Most Important)

Be ready with definitions + real examples.

- Class & Object
- Encapsulation
- Abstraction (Abstract class vs Interface)
- Inheritance (Method Overloading & Overriding)
- Polymorphism
- Access Modifiers
- this & super keyword
- Static vs Instance members
- Composition vs Aggregation
- Deep Copy vs Shallow Copy

>>>

Java Collections Framework

- List, Set, Map Interfaces
- ArrayList vs LinkedList
- HashSet, LinkedHashSet, TreeSet
- HashMap vs LinkedHashMap vs TreeMap
- Queue and PriorityQueue
- Comparator vs Comparable
- Fail Fast vs Fail Safe Iterators
- How HashMap works internally (**VERY IMPORTANT**)
- Big-O complexities of operations
- Concurrent Collections (ConcurrentHashMap, CopyOnWriteArrayList)

>>>

Exception Handling

- Checked vs Unchecked Exceptions
- try / catch / finally
- throw vs throws
- Custom Exceptions
- Error vs Exception

>>>>

Multithreading & Concurrency

- Thread lifecycle
- Creating threads: Thread, Runnable, Executors
- Synchronization
- Locks (ReentrantLock)
- Volatile Keyword
- Atomic Variables
- Executor Service, Thread Pools
- CompletableFuture
- Deadlock, Starvation, Race condition

>>>

Java Memory Management

- Stack vs Heap
- Garbage Collection (GC)
- Generations (Young, Old, Eden, Survivor)
- Soft, Weak, Phantom references
- Memory leaks
- JVM Tuning basics

>>>>

Java 8 and Functional Programming

- Lambda expressions
- Functional Interfaces (Supplier,
➤ Consumer, Predicate, Function)
- Streams API (map, filter, reduce,
collect)
- Optional Class
- Method references
- Default & Static methods in interfaces

>>>>

End of Part 1

Do follow For Part 2