30-Day Java Backend Interview Preparation Plan (30 LPA Target)

Daily Time Commitment: 1–1.5 hours

Daily Structure (≈75–90 minutes)

- Core Backend (Java / Spring): ~25 min
- DSA: ~25 min
- SQL / System Design / AI: ~20–30 min
- 5 min verbal recap

Day 1

Java: JVM vs JRE vs JDK, Bytecode, Program execution flow **DSA:** Arrays memory layout, static vs dynamic arrays **SQL:** SELECT, WHERE, ORDER BY, query execution order

Day 2

Java: JVM memory – Stack, Heap, Metaspace, object creation **DSA:** Prefix sum technique **SQL:** COUNT, SUM, AVG, NULL handling

Day 3

Java: OOP principles (encapsulation, abstraction enforcement) **DSA:** Two pointers technique **System Design:** Vertical vs horizontal scaling

Day 4

Java: equals() & hashCode() contracts **DSA:** Hashing fundamentals **SQL:** GROUP BY, HAVING

Day 5

Java: Collections overview (List, Set, Map) **DSA:** Sliding window (fixed size) **System Design:** Load balancers (L4 vs L7)

Day 6

Java: ArrayList internals, resizing **DSA:** Stack fundamentals **SQL:** INNER & LEFT JOIN

Day 7 (Revision)

Java: JVM + Collections recap **DSA:** Mixed problems **System Design:** Caching basics

---

Day 8

Spring: Why Spring, loose coupling **DSA:** Queue & BFS intuition **SQL:** RIGHT & FULL JOIN

Day 9

Spring: IOC container, DI types **DSA:** Linked List basics **SQL:** Subqueries & correlated subqueries

Day 10

Spring: Bean lifecycle, @Component vs @Bean **DSA:** Fast & slow pointer **System Design:** REST principles & idempotency

Day 11

Spring Boot: Auto-configuration, starters **DSA:** Linked list reversal **SQL:** Indexing & B-Tree basics

Day 12

Spring: REST controllers, request lifecycle **DSA:** Recursion basics **System Design:** Rate limiting strategies

Day 13

Spring: DTO vs Entity **DSA:** Binary search patterns **SQL:** ACID properties

Day 14 (Revision)

Spring: IOC + REST recap **DSA:** Timed practice **System Design:** Pagination strategies

---

Day 15

Java: HashMap internals **DSA:** Tree basics **SQL:** Query execution plan intro

Day 16

Java: ConcurrentHashMap **DSA:** Tree traversals **System Design:** Database schema design

Day 17

Java: Thread lifecycle **DSA:** BST operations **SQL:** Normalization

Day 18

Java: Synchronization & locks **DSA:** Heap / PriorityQueue **System Design:** Sharding vs replication

Day 19

Java: Executors framework **DSA:** Graph basics, BFS/DFS **SQL:** Transactions & isolation levels

Day 20

Java: Deadlocks & prevention **DSA:** Graph cycle detection **System Design:** Failure handling

Day 21 (Revision)

Java: Concurrency recap **DSA:** Medium problems **System Design:** Monitoring & logging

---

Day 22

Spring: JPA basics, entity lifecycle **DSA:** DP mindset **AI:** AI vs ML vs DL (backend view)

Day 23

Spring: Lazy vs Eager loading **DSA:** 1D DP **AI:** LLM basics, tokens

Day 24

Spring: N+1 problem **DSA:** DP patterns **AI:** Embeddings & vector DBs

Day 25

Spring: Transactions, propagation **DSA:** DP optimization **AI:** Using AI APIs in backend

Day 26

Spring: JWT auth, filters vs interceptors **DSA:** Revision **System Design:** URL Shortener

Day 27

Spring: Exception handling, @ControllerAdvice **DSA:** Mock problems **System Design:** Notification system

Day 28

Java: Garbage collectors, STW **DSA:** Weak areas **System Design:** Order management system

Day 29

Revision: Java + Spring deep revision **DSA:** Timed mock **System Design:** End-to-end design discussion

Day 30

Final: Resume walkthrough, project explanation, weak area cleanup

---

Outcome: Strong fundamentals, interview-ready explanations, system-level thinking.