Java Core — Most Important Concepts Only

This document focuses only on the highest value Java concepts that give the most practical industry impact. The goal is to avoid overload and focus on what is truly needed to write clean, modern Java code.

Stage 1 — OOP (Must Know)

• Classes and Objects — Blueprint vs real instance, object memory basics.

• Constructors — Ensure objects are valid when created, enforce required data.

• Encapsulation — Private fields with controlled access, protects data integrity.

• Getters (and limited Setters) — Read data safely, restrict modifications when needed.

• Inheritance (Basic) — Code reuse using extends and method overriding.

• Interfaces — Define contracts, used heavily in frameworks like Spring.

• Composition — Prefer HAS-A relationships instead of IS-A when possible.

• equals() and hashCode() — Required for correct behavior in Sets and Maps.

• Immutability and Records — Safer objects, modern Java data modeling.

Stage 2 — Functional Programming (Must Know)

• Lambdas — Short way to pass behavior and simplify code.

• Functional Interfaces — Predicate, Function, Consumer, Supplier.

• Streams — Focus on filter, map, forEach, collect(toList), findFirst.

• Optional — Safe null handling using ofNullable, ifPresent, orElse.

Stage 3 — Builder Pattern (Must Know)

• Why Builder Exists — Avoid constructors with too many parameters.

• Builder Structure — Inner Builder class, chain methods, build() method.

• When To Use Builder — Objects with many optional fields or complex setup.

Ultra Short Memory Version

OOP → Class, Constructor, Encapsulation, Interface, Composition, Record / Immutability
Functional → Lambda, Streams (filter, map, collect), Optional Design → Builder Pattern