Swiggy – Software Development Engineer (SDE-1)

Low-Level Design Interview Question

Design an **in-memory Online E-Commerce Cart System** using **Core Java only**. This exercise evaluates object-oriented design, data modeling, and clean code practices. No database or external APIs are required.

Time Expectation: 45–60 minutes

Difficulty Level: SDE-1

Problem Constraints & Guidelines

- Each user has exactly one cart.
- The system must be fully in-memory.
- Use only Core Java (classes, interfaces, enums, collections, exceptions).
- No frameworks, no databases, no REST or UI layers.
- Focus on clean object-oriented design and extensibility.
- Thread safety is not mandatory unless explicitly handled by the candidate.
- Code should be readable and logically structured.

Task 1: Cart Ownership & Basic Add

Design the cart such that it belongs to a single user identified by a userId. The user should be able to add products to the cart by providing productId and quantity. Define how the cart behaves when the same product is added multiple times. Quantity values must be validated according to reasonable business rules.

Task 2: Update Quantity & Remove Product

Support updating the quantity of a product already present in the cart. Allow removing a product entirely from the cart. Ensure invalid operations are handled gracefully, such as updating or removing products that do not exist in the cart or setting invalid quantities.

Task 3: Product Variants & Cart Line Uniqueness

Support product variants such as size, color, or storage. A cart line should be uniquely identified by a combination of product and its variant. Different variants of the same product must be treated as separate cart entries. Adding a product with the same variant again should follow consistent cart behavior.

Task 4: Pricing & Cart Summary

Calculate pricing details for the cart including line totals, subtotal, tax, shipping fee, and final payable amount. Tax and shipping rules can be assumed to be simple and configurable. The cart summary should be printable in a clean, readable format.

Task 5: Coupons & Discounts

Support applying a single coupon to the cart. Coupons may represent flat discounts or percentage-based discounts with caps. Ensure discounts never make the payable amount negative. Coupons should be replaceable or removable. Invalid or non-applicable coupons must be handled correctly.

Note: Candidates may write a simple main() method to demonstrate cart behavior.