

* Benefits of OOD ->

1) Encapsulation

3) Object

5) Data hiding

7) Polymorphism

2) Class

4) Data Abstraction

6) Inheritance

8) Message Passing.

* Project Scenario :- Just wondering to develop an application for my father's business, i.e. - "A Cloth Store".

Key features -> A cloth store - has products to sales, discounts, GST & Taxes Methods, Customer of Types < Wholesale OR, RETAIL.

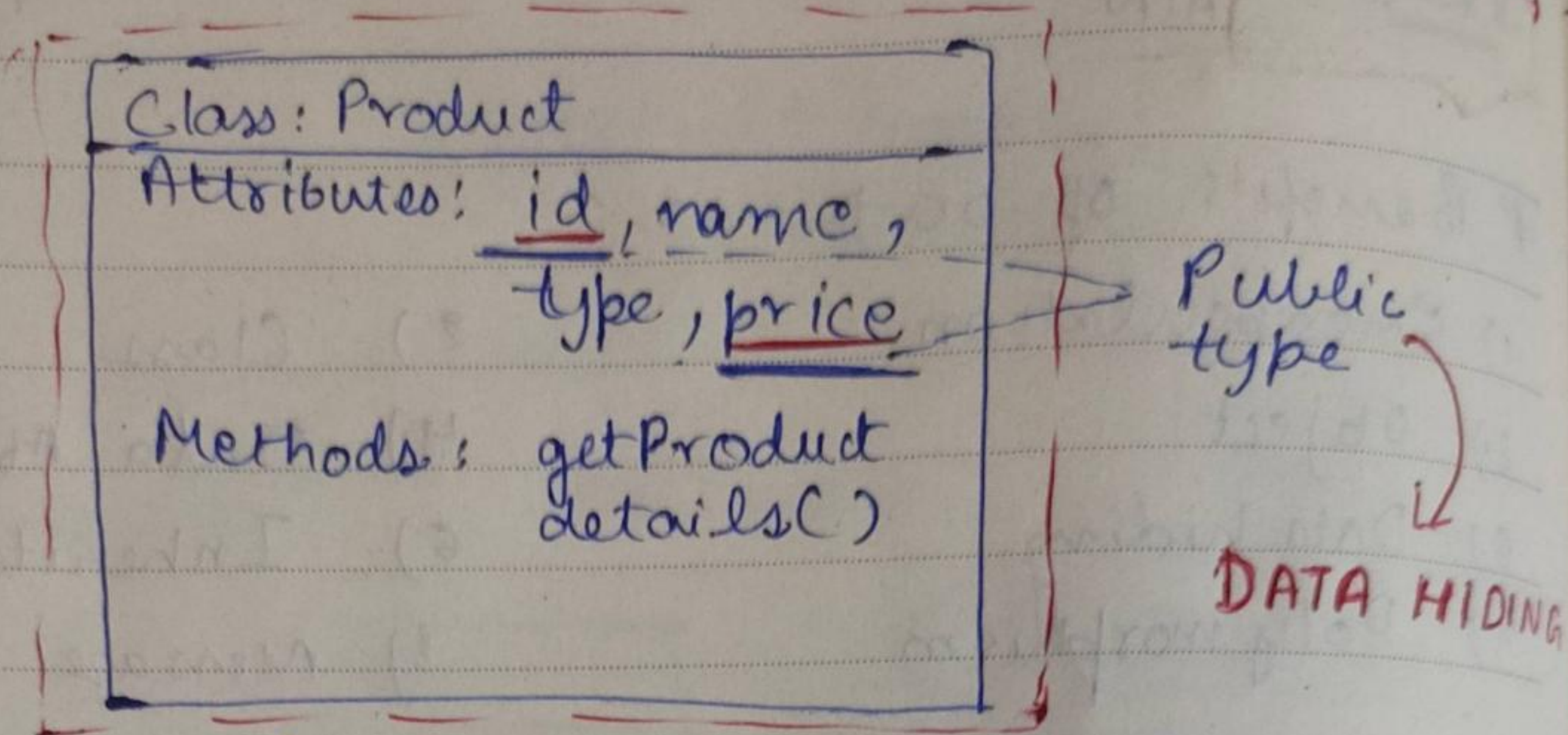
Price calculation methods etc.

So, when we think of these functionalities Object oriented designing is the best way to implement them!

Classes -> Product, Customer :- Retailer, Wholesaler

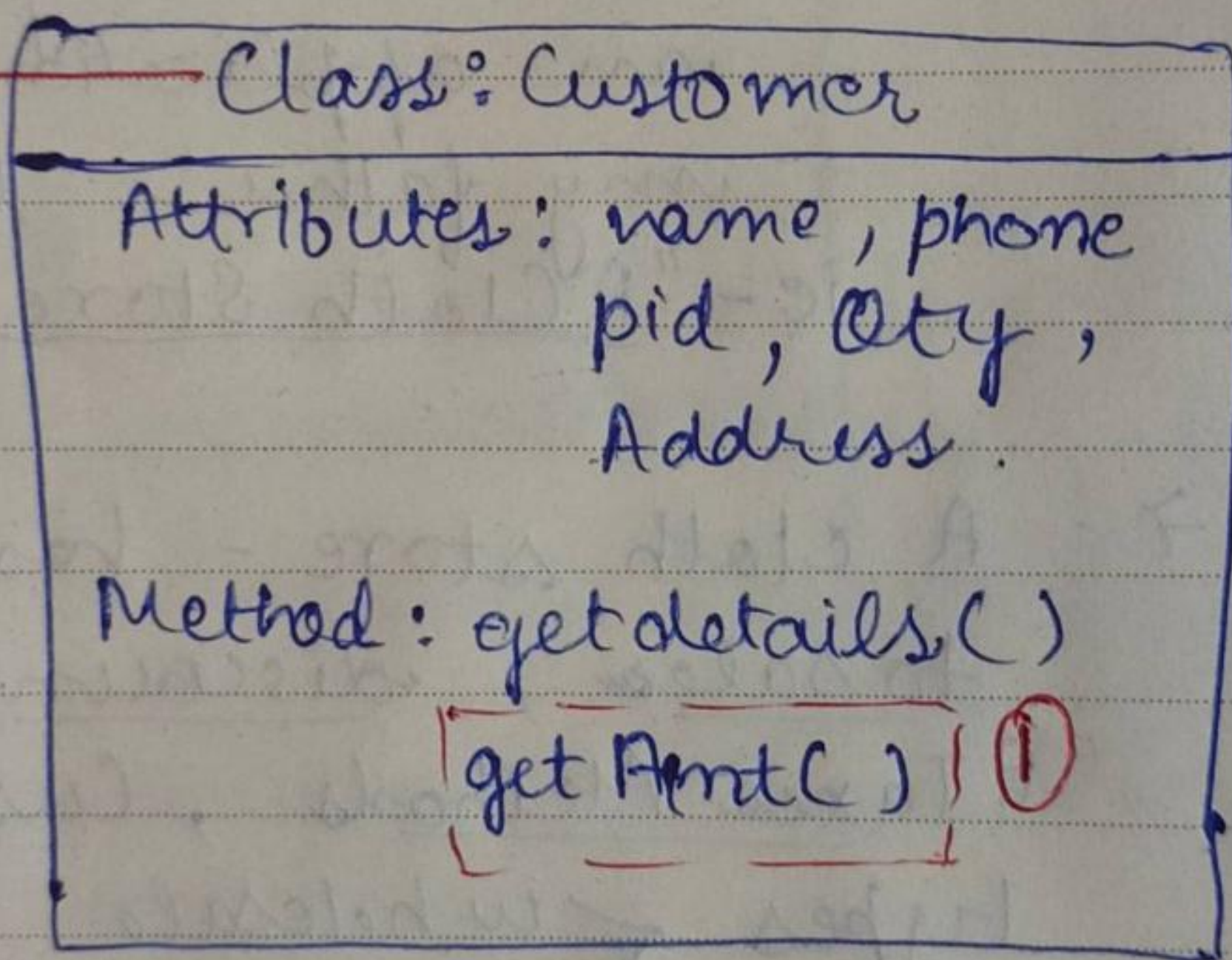
Methods like - getProductDetails(), getAmt()
(with or without taxes), etc

ENCAPSULATION



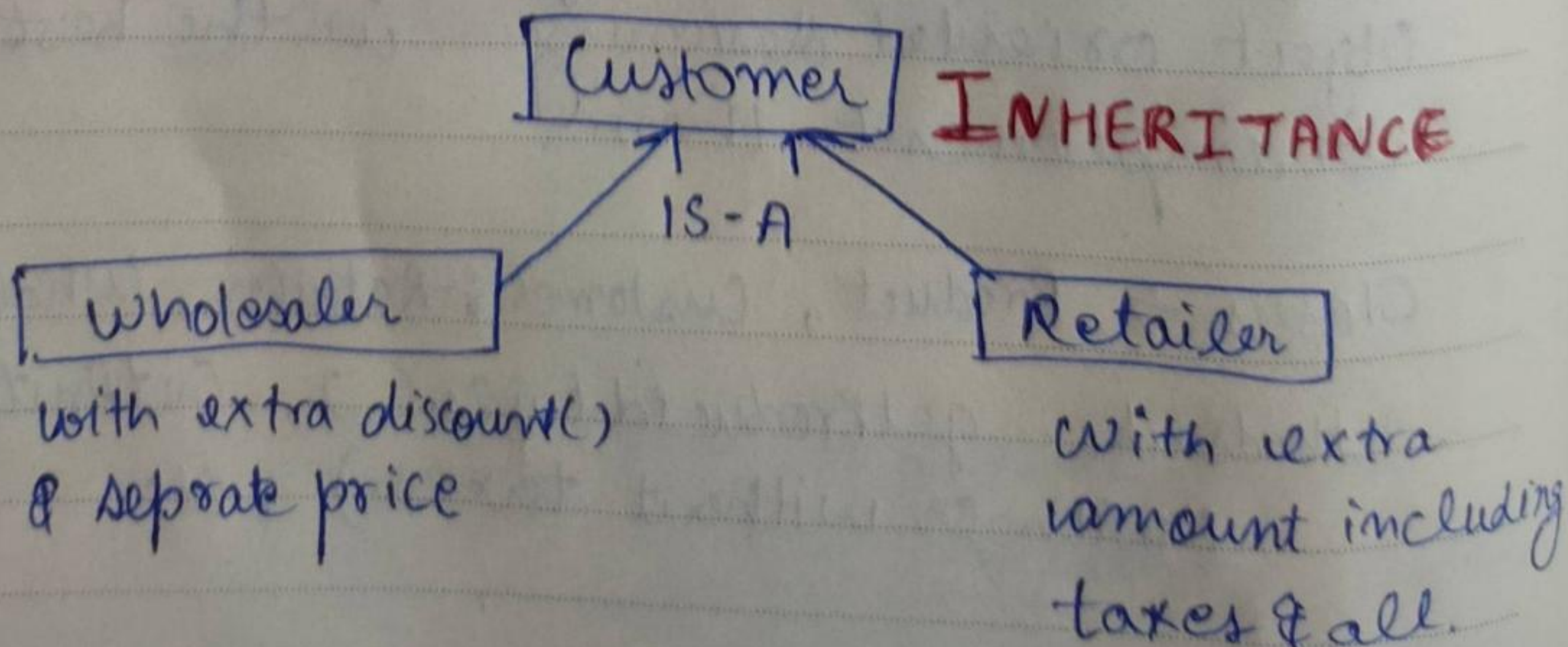
Product As Object

CLASS



Customer As Object

// which can be further divided //



in a "CLOTH STORE" Application

Class: Wholesaler

Attributes: Like
Parent Class 'Customer'

Methods: LessGST(), etc.

discount(COD, GST, CreditNote)

② Get ~~Total~~ Amt() (After deducting discounts)

Class: Retailer

Attributes: Same like
Parent Class 'Customer'

Methods:

AddGST()

AddProfit()

③ Get ~~Total~~ Amt()

① Get Amt()

② Get Amt (GST, Discount)

③ Get Amt (GST, ST, ProfitPercent)

} POLYM-
ORPHISM