

# Lecture 24:- Build Discount Coupon System.

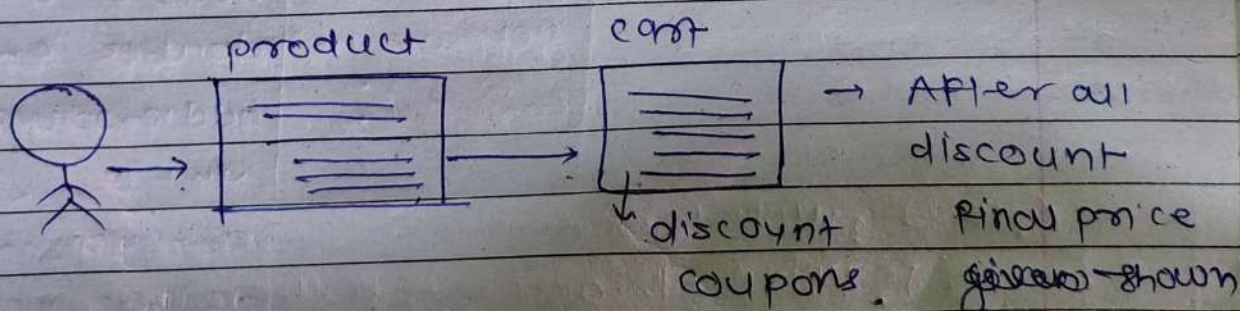
Page No.

Date

## # functional Requirements:-

- ① we can add New coupons at runtime
- ② Both cart and product level discounts.
- ③ support multiple coupon types.  
(seasonal, loyalty discount, banking discount).
- ④ support flat and percentage discount both.
- ⑤ one coupon can / can't be applied on the top of other coupons.
- ⑥ The real-time

## # Happy Flow



- plug and play model.

- no need to make user class

as it will be handled by optimization application which will integrate them.



we make strategy-pattern Manager  
(factory-pattern) which-talk with  
diff-strategies.

<<enum>>	
type	
FLAT;	
PERCENT;	
PERCENTWITHCAP;	

strategy manager
get coupon start (type)

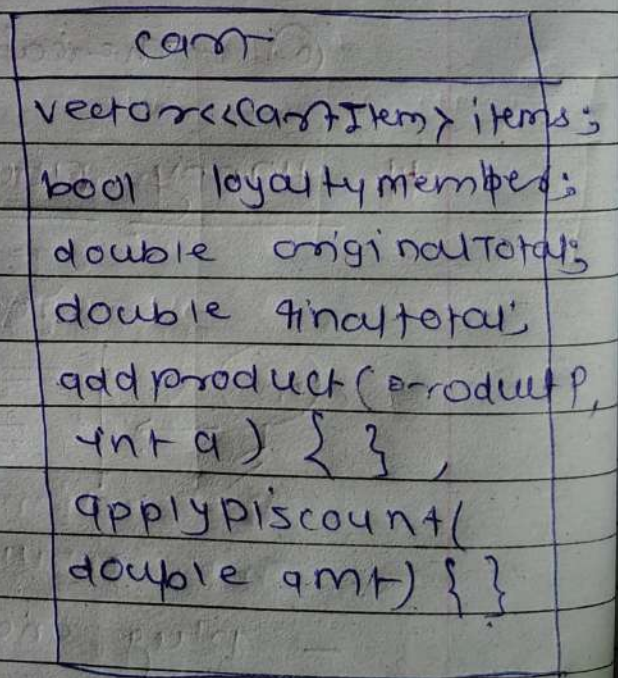
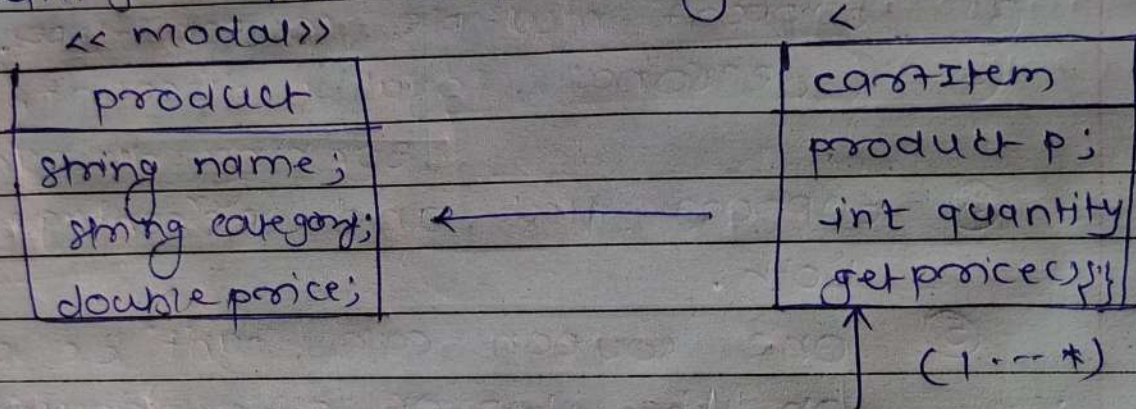
first point of contact for coupon:-

thread safe	coupon manager	
	coupon * head;	
	mutex mtx;	(1...*)
	register coupon(coupon)	→ coupon
	ApplyAll coupon (head) {}	
	get coupon applicable (coupon) {};	



## # UML Diagram

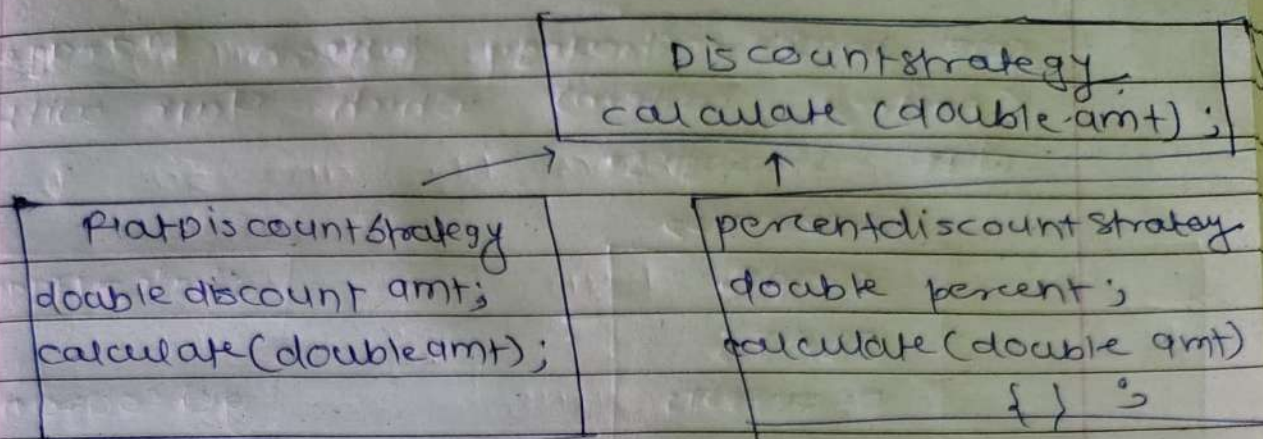
we will create generic product class as this class does not directly link with this application because we don't know where we are going to integrate this coupon engine.



Now, we will use

strategy dp to create diff types of discounts like flat %, client can use alternatively this strategy





- Let's create our coupon class.  
coupon:- coupon can't be applied  
on top of other coupons.

To Achieve this we have two patterns:-

1. Decorator
2. Chain of Responsibility

