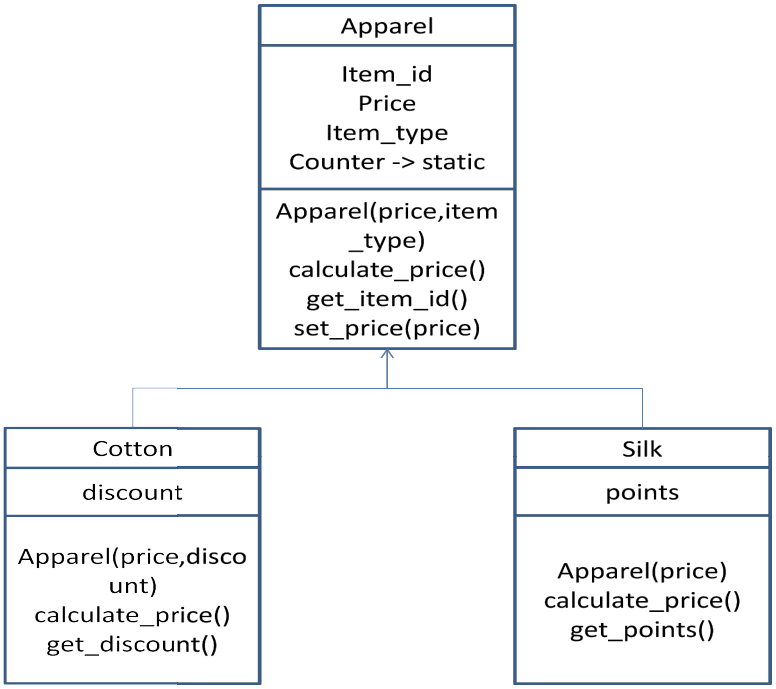
1. An apparel shop wants to manage the items which it sells. Write a JAVA program to implement the class diagram given below.



Class Description:
Apparel class:

1. Initialize static variable counter to 100
2. In the constructor, auto-generate item_id starting from 101 prefixed by "C" for cotton apparels and "S" for silk apparels. Example – C101, S102, S103, C104 etc.
3. **calculate_price():** Add 5% service tax on the price of the apparel and update attribute, price with the new value

Cotton class:

1. While invoking parent constructor from child constructor, pass "Cotton" as item_type
2. **calculate_price():** Update attribute, price of Apparel class based on rules given below
    a. Add service tax on price by invoking appropriate method of Apparel class
    b. Apply discount on price
    c. Add 5% VAT on final price

Silk class:

1. While invoking parent constructor from child constructor, pass "Silk" as item_type
2. **calculate_price():** Update attribute, price of Apparel class based on rules given below
    a. Add service tax on price by invoking appropriate method of Apparel class

b. Identify points earned based on rules given below:

Silk apparels with price more than Rs. 10000, earn 10 points and anything less than or equal to that earn 3 points

c. Initialize attribute, points with the identified points
d. Add 10% VAT on price

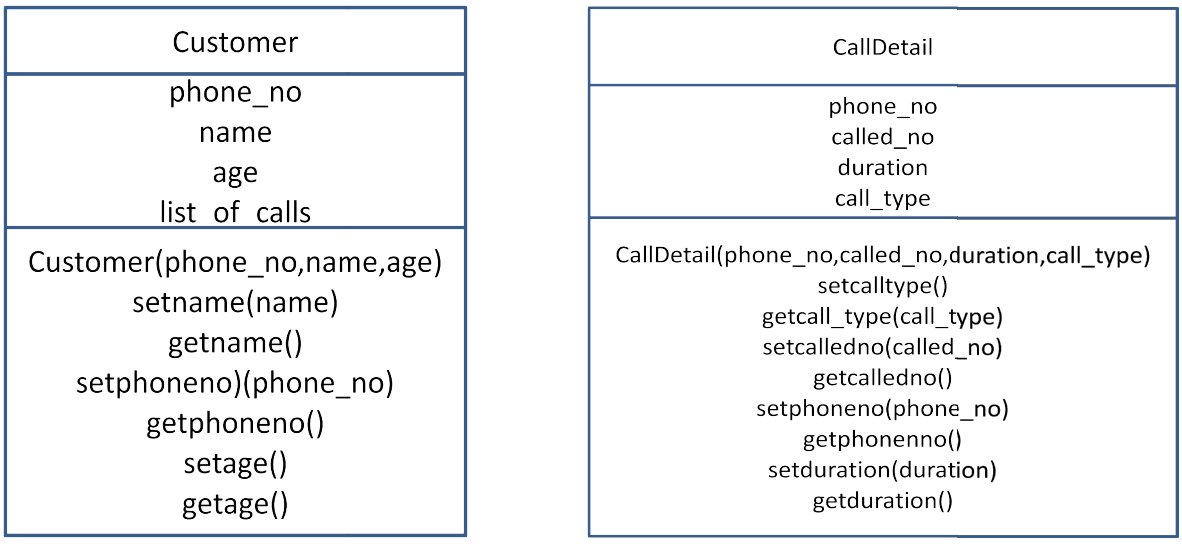
Note: Perform case sensitive string comparison .

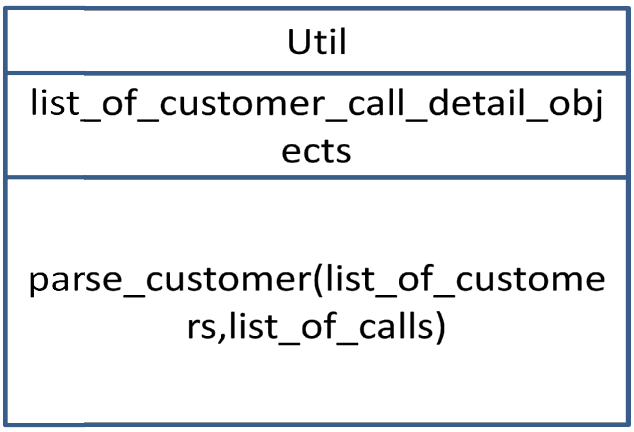
For testing:

- Create objects of Cotton class and Silk class
- Invoke **calculate_price()** on Cotton objects and Silk objects
- Display their details

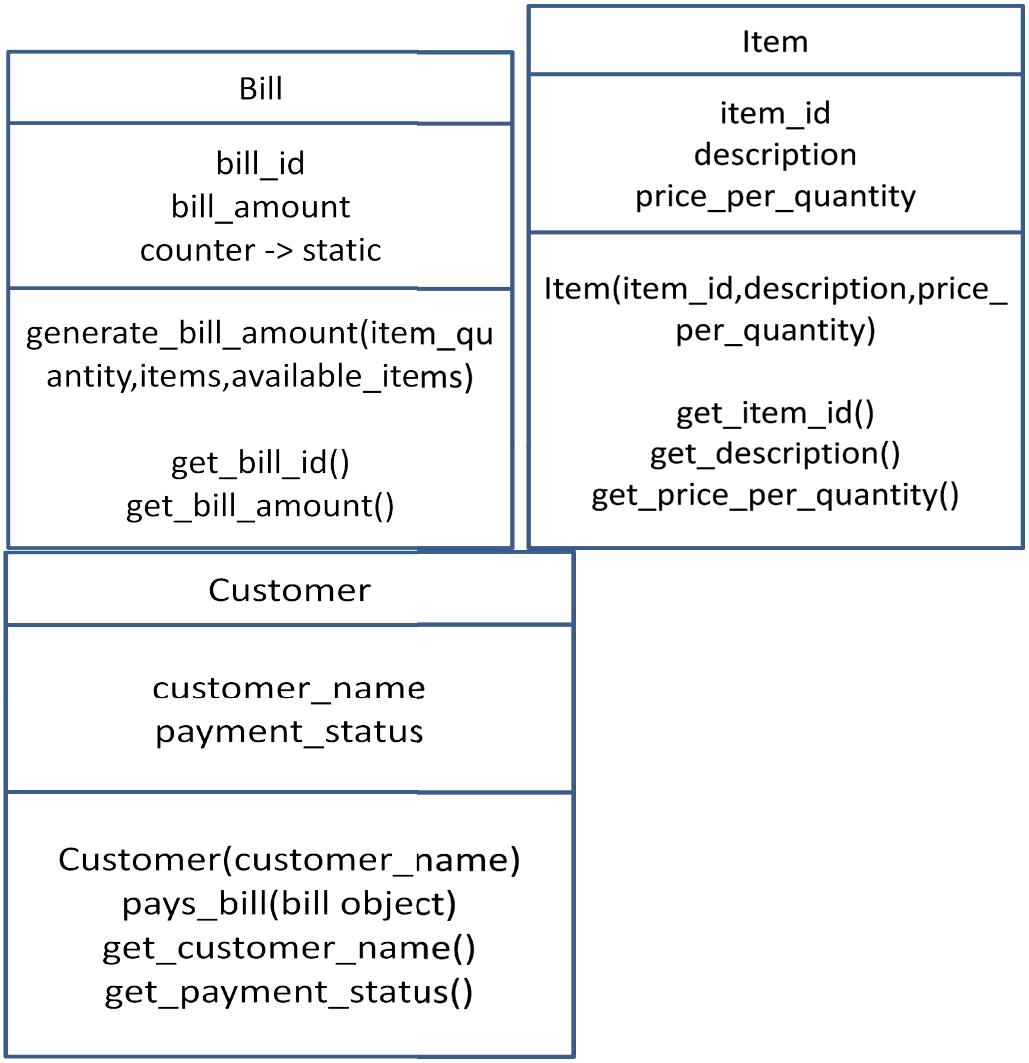
2. A telecom company wants to generate reports on the call details of the customers. Each customer can make multiple phone calls.

Class Diagram Description: The parse_customer method takes a list of Customer objects and a list of CallDetail objects. For every customer, identify all the corresponding Call Detail objects (the customer phone number and the phone number of Call detail object have to match), add them to a list and assign that list to the corresponding customer object.





3. In the retail store scenario, let's look at the portion of customer purchasing items from the retail store. Write a JAVA program to implement the class diagram given below.



Class Description:
Bill class:

1. Initialize static variable counter to 1000
2. **generate_bill_amount(item_quantity,items,available_items):** Calculate bill amount based on the items purchased by the customer
3. Accept list for item_quantity and items, first index item in list purchased by customer is in quantity is defined at first index of item_quantity list.
4. Accept a list, available_items which contains the list of Item objects available in the store
5. Generate bill id starting from 1001 prefixed by "B" and initialize attribute, bill_id. Ex. "B1001", "B1002" etc.
6. Calculate bill amount based on the quantity and price of the items purchased by the customer
7. Set attribute, bill_amount with the calculated bill amount

Assume that values in item_quantity and items are always valid.

Customer class:

pays_bill(bill): Pay bill based on the bill amount

a. Accept Bill object which contains the details of the bill to be paid by the customer
b. Update attribute, payment_status to "Paid"
c. Display customer name, bill id and bill amount

For testing:

- Create objects of Customer class, Item class and Bill class
- Invoke generate_bill_amount(item_quantity,items,available_items) on Bill object by passing the Lists containing item_quantity, items and available_items purchased by the Customer.
- Invoke pays_bill() on Customer object by passing the Bill object