1. Complete the Item class to make the program runnable. The instance variables of Item should at least include title, (original) unit price, and quantity. Also write a test program with appropriate test cases.

* I added a New Item class with all private instance variables (i.e. \_title, \_unitPriceOrgi, \_quantity with appropriate datatype) and getter and setter where needed for them to **encapsulate** those private variables from external classes. Also, I added new Test package and appropriate jUnit Test Class with test cases.

1. Update the Sale class to support different discounts for seniors and preferred customers. Update your test program with appropriate new test cases. Your program must use switch (case) statement(s).

* I created an Enumeration Class to check the Customer Type and based on that I **switch** over the Condition to check and verify applicable the discount rates.

Also, I added Test class and cases for calculation and methods.

1. Update your program for (2) to support that the senior’s discount is only on Tuesday. Update your test program with appropriate new test cases. Your program must keep the switch (case) statement(s).

* I checked the current day of the week and compare it with today’s name. Then, based on that condition I make assumption that it will give **0.10** discount rate that day for **SENIORS** members only whereas by default **0.05** for other days and other customer types.

1. Refactor your program for (3) to eliminate the switch statement(s) by using polymorphism, and retest your program.

* I created a new **Customer Class** and separate the **Calculation of the discount rate** based on **CustomerType** and day of the week part from Sales Class on this new Customer Class.

1. **(CS 571 Graduate Students Only)** Update your program for (4) to handle the new discount policy “buy one, get the second (and the rest of the same item) 50% off” for certain items.

* I checked the number of items using count to check whether the same item name exist on the added item list of the customer’s cart or not and then for every items valid for this promotion, after second item it gives a discount rate of 0.5(half) from the original price.

For example, purpose for now I did allow this rule to “**iPhone**” item. So if cart contains more than 2 items it will give discount of half of original price for every iPhone after 1st count.