Program: Online shopping cart (Part 2)

*Note: Creating multiple Scanner objects for the same input stream yields unexpected behavior. Thus, good practice is to use a single Scanner object for reading input from System.in. That Scanner object can be passed as an argument to any methods that read input.*

This program extends the earlier "Online shopping cart" program. (Consider first saving your earlier program).

(1) Extend the ItemToPurchase class per the following specifications:

* Private fields
  + string itemDescription - Initialized in default constructor to "none"
* Parameterized constructor to assign item name, item description, item price, and item quantity (default values of 0). (1 pt)
* Public member methods
  + setDescription() mutator & getDescription() accessor (2 pts)
  + printItemCost() - Outputs the item name followed by the quantity, price, and subtotal
  + printItemDescription() - Outputs the item name and description

Ex. of printItemCost() output:

Bottled Water 10 @ $1 = $10

Ex. of printItemDescription() output:

Bottled Water: Deer Park, 12 oz.

(2) Create two new files:

* ShoppingCart.java - Class definition
* ShoppingCartManager.java - Contains main() method

Build the ShoppingCart class with the following specifications. Note: Some can be method stubs (empty methods) initially, to be completed in later steps.

* Private fields
  + String customerName - Initialized in default constructor to "none"
  + String currentDate - Initialized in default constructor to "January 1, 2016"
  + ArrayList cartItems
* Default constructor
* Parameterized constructor which takes the customer name and date as parameters (1 pt)
* Public member methods
  + getCustomerName() accessor (1 pt)
  + getDate() accessor (1 pt)
  + addItem()
    - Adds an item to cartItems array. Has parameter ItemToPurchase. Does not return anything.
  + removeItem()
    - Removes item from cartItems array. Has a string (an item's name) parameter. Does not return anything.
    - If item name cannot be found, output this message: Item not found in cart. Nothing removed.
  + modifyItem()
    - Modifies an item's description, price, and/or quantity. Has parameter ItemToPurchase. Does not return anything.
    - If item can be found (by name) in cart, check if parameter has default values for description, price, and quantity. If not, modify item in cart.
    - If item cannot be found (by name) in cart, output this message: Item not found in cart. Nothing modified.
  + getNumItemsInCart() (2 pts)
    - Returns quantity of all items in cart. Has no parameters.
  + getCostOfCart() (2 pts)
    - Determines and returns the total cost of items in cart. Has no parameters.
  + printTotal()
    - Outputs total of objects in cart.
    - If cart is empty, output this message: SHOPPING CART IS EMPTY
  + printDescriptions()
    - Outputs each item's description.

Ex. of printTotal() output:

John Doe's Shopping Cart - February 1, 2016

Number of Items: 8

Nike Romaleos 2 @ $189 = $378

Chocolate Chips 5 @ $3 = $15

Powerbeats 2 Headphones 1 @ $128 = $128

Total: $521

Ex. of printDescriptions() output:

John Doe's Shopping Cart - February 1, 2016

Item Descriptions

Nike Romaleos: Volt color, Weightlifting shoes

Chocolate Chips: Semi-sweet

Powerbeats 2 Headphones: Bluetooth headphones

(3) In main(), prompt the user for a customer's name and today's date. Output the name and date. Create an object of type ShoppingCart. (1 pt)  
  
Ex.

Enter Customer's Name:

John Doe

Enter Today's Date:

February 1, 2016

Customer Name: John Doe

Today's Date: February 1, 2016

(4) Implement the printMenu() method. printMenu() has a ShoppingCart parameter, and outputs a menu of options to manipulate the shopping cart. Each option is represented by a single character. Build and output the menu within the method.

If the an invalid character is entered, continue to prompt for a valid choice. *Hint: Implement Quit before implementing other options.* Call printMenu() in the main() method. Continue to execute the menu until the user enters q to Quit. (3 pts)  
  
Ex:

MENU

a - Add item to cart

d - Remove item from cart

c - Change item quantity

i - Output items' descriptions

o - Output shopping cart

q - Quit

Choose an option:

(5) Implement Output shopping cart menu option. (3 pts)  
  
Ex:

OUTPUT SHOPPING CART

John Doe's Shopping Cart - February 1, 2016

Number of Items: 8

Nike Romaleos 2 @ $189 = $378

Chocolate Chips 5 @ $3 = $15

Powerbeats 2 Headphones 1 @ $128 = $128

Total: $521

(6) Implement Output item's description menu option. (2 pts)  
  
Ex.

OUTPUT ITEMS' DESCRIPTIONS

John Doe's Shopping Cart - February 1, 2016

Item Descriptions

Nike Romaleos: Volt color, Weightlifting shoes

Chocolate Chips: Semi-sweet

Powerbeats 2 Headphones: Bluetooth headphones

(7) Implement Add item to cart menu option. (3 pts)  
  
Ex:

ADD ITEM TO CART

Enter the item name:

Nike Romaleos

Enter the item description:

Volt color, Weightlifting shoes

Enter the item price:

189

Enter the item quantity:

2

(8) Implement Remove item menu option. (4 pts)  
  
Ex:

REMOVE ITEM FROM CART

Enter name of item to remove:

Chocolate Chips

(9) Implement Change item quantity menu option. *Hint: Make new ItemToPurchase object and use ItemToPurchase modifiers before using modifyItem() method.* (5 pts)  
  
Ex:

CHANGE ITEM QUANTITY

Enter the item name:

Nike Romaleos

Enter the new quantity:

3