* (1) Extend the ItemToPurchase class per the following specifications:
* Parameterized constructor to assign item name, item description, item price, and item quantity (default values of 0). (1 pt)
* Public member functions
* SetDescription() & GetDescription() (2 pts)
* PrintItemCost() - Outputs the item name followed by the quantity, price, and subtotal
* PrintItemDescription() - Outputs the item name and description
* Private data members
* string itemDescription - Initialized in default constructor to "none"
* *Ex*. of PrintItemCost() output:

Bottled Water 10 @ $1 = $10

* *Ex*. of PrintItemDescription() output:

Bottled Water: Deer Park, 12 oz.

* (2) Create three new files:
* ShoppingCart.h - Class declaration
* ShoppingCart.cpp - Class definition
* main.cpp - main() function (Note: main()'s functionality differs from the warm up)
* Build the ShoppingCart class with the following specifications. Note: Some can be function stubs (empty functions) initially, to be completed in later steps.
* Default constructor
* Parameterized constructor which takes the customer name and date as parameters (1 pt)
* Private data members
* string customerName - Initialized in default constructor to "none"
* string currentDate - Initialized in default constructor to "January 1, 2016"
* vector < ItemToPurchase > cartItems
* Public member functions
* GetCustomerName() accessor (1 pt)
* GetDate() accessor (1 pt)
* AddItem()
  + Adds an item to cartItems vector. Has parameter ItemToPurchase. Does not return anything.
* RemoveItem()
  + Removes item from cartItems vector. 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() function. 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 function.
* 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() function. 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() function.* (5 pts)   
    
  *Ex*:

CHANGE ITEM QUANTITY

Enter the item name:

Nike Romaleos

Enter the new quantity:

3