# Assignment 2 E-commerce

Due Date: 11:59 pm on September 22 Extended, Wednesday

## Objectives:

1. Students will use class inheritance to solve a problem
2. Students will write correct, will-documented and readable programs in a reasonable amount of time.

## Problem Description:

When shopping online, you select items and add them to a shopping cart. Duplicate items are permitted in a shopping cart, as you can purchase multiples of the same item. You also can remove an item from a shopping cart, if you change your mind about buying it. The shopping cart can show its current contents with their prices and the total cost of these items. Design the ADT item and shopping cart.

### Task 1 – Item (a regular class, not a template)

- Declare three data members: *name of item*, *unit price of item*, and *quantity of item*
- Define a default constructor
- Define a constructor that initializes all three data members by the parameters.
- Define ***set*** and **get** functions for each data member.
- Define the following operators:
  Equal to: ==
  two items are same if their name and price are same; quantity is required to be same too!
  Input : >>
  Output: <<

**!!Test the class Item before you move to next task!!**

### Task 2 - Shopping cart (a regular class, not a template)

- Design the shopping cart as a derived class from the Bag class using public inheritance as follows:
  ```
  class ShoppingCart:public Bag<Item>
  ```
- Declare *total price* as a data member.
- Define a default constructor.
- Define a member function to *get* the total price.
- Override the base class methods: *add*, *remove* and *clear*.
  [hint] Similar to *setItem()* function in *toyBox* class

**!!Test the class before you move to next task!!**

## Requirements

- Write and document the class in **header and implementation files**, and compile it separately from the client program.
- Add the implementation of ShoppingCart class at the end of the header file “ShoppingCart.h” as follows:

```
#include "ShoppingCart.cpp"
```

(before the line `#endif`)

**This is required to run your program correctly on GitHub.

## Testing

Design your test data to test ALL operations provided in the ShoppingCart:

- Add
- Remove ( two cases: in the cart and not in the cart)
- Change quantity (two cases: in the cart and not in the cart). If in the cart, validate the quantity.

The running result of the program must look like this:

```
g++ project2.cpp
a.out

Welcome to XXX SHOPPING CENTER.
Enter the item you selected as the following order:
name unitPrice quantity
(Name can not contain any space. Otherwise errors happen!)
--> T-shirt 19.99 2
Want to continue y/n-->y

-->Sweater 39.99 2
Want to continue y/n-->y

-->iphone_case 25.50 3
Want to continue y/n-->y

-->Towel 9.99 5
Want to continue y/n-->n

Here is your order:
-------------------------------------------------
Name                Unit_Price          Quantity
T-shirt             $19.99               2
Sweater             $39.99               2
iphone_case         $25.5                3
Towel               $9.99                5
The total charge is $246.41
--------------------------------------------------

Want to modify your order? y/n-->y
What do you want? Enter 1: add 2: remove 3: change quantity
--> 1
Enter the item to add as the following order:
name unitPrice quantity
```

```
--> shoe 99.99 1
The item has been added.

Want to modify your order? y/n-->y
What do you want? Enter 1: add 2: remove 3: change quantity
--> 2
Enter the item to remove as the following order:
name unitPrice quantity
--> Sweater 29.99 2
No such item in your shopping cart!

Want to modify your order? y/n-->y
What do you want? Enter 1: add 2: remove 3: change quantity
--> 2
Enter the item to remove as the following order:
name unitPrice quantity
--> Sweater 39.99 2
The item has been removed.

Want to modify your order? y/n-->y
What do you want? Enter 1: add 2: remove 3: change quantity
--> 3
Enter the item to change as the following order:
name unitPrice quantity
--> Towel 9.99 5
Enter a new quantity --> -2
-2 is not a valid input.
Enter a new quantity --> 2
The quantity has been modified.

Want to modify your order? y/n-->n


Here is your updated order:
-------------------------------------------------
You have ordered the following items:
Name                Unit_Price          Quantity
T-shirt             $19.99               2
Shoe                $99.99               1
iphone_case         $25.5                3
Towel               $9.99                2
The total charge is $236.45
Thanks for shopping in XXX SHOPPING CENTER.
---------------------------------------------------
```

## Other requirements

- For each program, add the following information at the top of the file:

  - Description of the problem to solve
  - Your startID and name
  - Due Date
  - Instructor
  - Comments
- For Classes and Functions, add necessary documentation and comments using **javadoc-style** comment. For more information about Javadoc style comment, please refer to Appendix C: C++ Documentation Systems from the textbook. You can also follow the document About Programming Assignments posted on D2L.

## What to Hand In

- Submit all of the source programs to your class account in **GitHub** and test well.
- Submit the following documents to the drop box **Project2 on D2L**:
  - *bag.h*, *bag.cpp* and *bagInterface.h*
  - *item.h* and *item.cpp*
  - *shoppingCart.h* and *shoppingCart.cpp*
  - the application program that simulates online shopping cart
  - a script file for test runs on **GitHub.**
    - Don't list your program source code in the script file!!!
  - a word file that contains
    - the design document that includes
      - a simple UML diagram for ***Item*** class
      - a simple UML diagram for ***shoppingCart*** class

      ** UML diagram is covered in Chapter 13 of CSCI 201 textbook. The slides of this chapter are posted on D2L.
    - a user document that contains the following sections:
      - description about the problem to solve
      - file location on GitHub
      - how to compile the program
      - how to run the program

  (Please follow the details of the document About Programming Assignments posted on D2L.)

| Requirements | points |
|---|---|
| Comments in the program | 20 |
| Program correctness | 30 |
| Script file from several test runs | 30 |
| Design document | 10 |
| User document | 10 |
| **TOTAL POINTS** | **100** |

No Submission Made

| Requirements | points |
|---|---|
| Comments in the program | 0 |
| Program correctness | 0 |
| Script file from several test runs | 0 |
| Design document | 0 |
| User document | 0 |
| **TOTAL POINTS** | **0** |