CSE341 – File Organization

Fall 2019, Lab Assignment-1

Write a C++ program which will provide a shopping list of products to users. The users should be able to create a shopping list, insert a product to an existing shopping list, check a product when it is bought, print the entire shopping list, print the unbought products in the shopping list.

You must implement 2 classes (ShoppingList and Product). Product class must have 5 attributes (int id, string name, int amount, bool priority(True if high/False if low) and bool bought(True if +/False if -)). ShoppingList class must have 3 attributes (int id, int size and vector of products). ShoppingList class must have 4 methods (insert, printAll, printShoppingList and checkBought).

insert(Product p): Insert method must take a product object as argument and add this product to the shopping list. If a product with same id exists in the shopping list, then print an error message "There is a product with the same id in the shopping list" and do not insert the product. If the size of shopping list has reached its limit, then print an error message "The shopping list is full"

printAll(): printAll method must print all the products in the shoppingList with their id, name, amount, priority and bought(- if not bought(default), + if bought). The method should print the products with high priority first.

printShoppingList(): printShoppingList method must print the products which are unbought. The printing format is the same as printAll method (id, name, amount, priority, bought). The method should print the products with high priority first.

checkBought(int id): checkBought method must take id of a product as an argument, update the product having that id by setting its bought attribute. If there is no product having this id, then print an error message "There is no product with that id".

You must use vectors to represent the shopping list. You must use vector iterator approach (not at() or array approach) in printAll, printShoppingList and checkBought methods.

You do not need to provide a menu for the user. Just implement the methods so that they can be called and executed like in the example in the second page. Your main.cpp will be your testing file.

Compress your main.cpp, shopping.h, shopping.cpp into a single zip file and name it in the format "NameLastnameA1.zip". Ex: "GokhanAkgunA1.zip". Make sure that, your code works properly when compiled with g++.

Example:

```
ShoppingList sl1(1,20);
Product p1(1, “bread”, 2, True);
Product p2(2, “egg”, 6, True);
Product p3(3, “milk”, 1, False);
Product p4(4, “water”, 1, True);
Product p5(5, “soda”, 3, False);
Product p6(3, “pasta”, 1, False);
sl1.insert(p1);
sl1.insert(p2);
sl1.insert(p3);
sl1.insert(p4);
sl1.insert(p5);
sl1.insert(p6);
        There is a product with the same id in the shopping list.
sl1.printAll();
        id      name            amount          priority        bought
        1       bread           2               high            -
        2       egg             6               high            -
        4       water           1               high            -
        3       milk            1               low             -
        5       soda            3               low             -
sl1.checkBought(4);
sl1.checkBought(6);
        There is no product with that id.
sl1.printAll();
        id      name            amount          priority        bought
        1       bread           2               high            -
        2       egg             6               high            -
        4       water           1               high            +
        3       milk            1               low             -
        5       soda            3               low             -
sl1.printShoppingList();
        id      name            amount          priority        bought
        1       bread           2               high            -
        2       egg             6               high            -
        3       milk            1               low             -
        5       soda            3               low             -
```