Project 5 – Groceries part 2

EGRE246 – Fall 2017

Assigned 10/25/2017 – Due 11/08/2017

In this project, we are going to apply Inheritance and Polymorphism to the grocery store. Unlike the last project involving groceries, we will not use linked lists. Also, unlike project 3, we will not make a priority queue, we are just interested in learning on how to make derived classes and learning how to utilize polymorphism to display what we want. This project will also focus on command line arguments for programming as well as reading from a file. Here is what you need to do:

Base class : item.h

Create a class called item. Each item should have private members:

- Name (string)
- Price (double)
- Bar Code Number (string – the number overflows integer and long long types)
- Classification (string)

Each item should have (at a minimum) public members:

- Constructors (default and overloaded) – initialize name, price, bar code, and classification. The classification should always be "general" in the item class (despite overloaded constructor).
- Set Item method – that sets the name, price, and bar code of the item. The classification shouldn't change in the set method.
- Display method – displays the name of the item, the classification, the price, and the barcode

Derived class : fruit.h

Create a class called fruit that inherits from the item class. Each fruit item should have private members:

- Sell by date's month (int)
- Sell by date's day (int)
- Sell by date's year(int)

Each item should have (at a minimum) public members:

- Constructors (default and overloaded) – initialize name, price, bar code, classification, and sell by date members. The classification should always be "fruit" in the item class (despite overloaded constructor). NOTE: You may need to use the Set Item method from item class to set some members.
- Set Item method – that sets the name, price, bar code, and sell by date members. The classification shouldn't change in the set method. NOTE: You may need to use the Set Item method from item class to set some members.
- Display method – displays the name of the item, the classification, the price, the barcode number, and the sell by date. NOTE: You may need to use the display method from item class.

Derived class : dairy.h

Create a class called dairy that inherits from the item class. Each dairy item should have private members:

- Expiration date's month (int)
- Expiration date's day (int)
- Expiration date's year(int)
- The storage temperature (double)

Each item should have (at a minimum) public members:

- Constructors (default and overloaded) – initialize name, price, bar code, classification, expiration date members, and storage temperature. The classification should always be "dairy" in the item class (despite overloaded constructor). NOTE: You may need to use the Set Item method from item class to set some members.
- Set Item method – that sets the name, price, bar code, expiration date members, and storage temperature. The classification shouldn't change in the set method. NOTE: You may need to use the Set Item method from item class to set some members.
- Display method – displays the name of the item, the classification, the price, the barcode number, and the sell by date. NOTE: You may need to use the display method from item class.

main.cpp

1) In main, you need to accept a file name as an argument.
2) Main should be able to tell if there is not enough arguments or too many arguments.
    a. If there are too few arguments, an error message should be displayed and the program should end.
    b. If there are too many arguments, an error message should be displayed and the program should end.
    c. Otherwise, the program should continue.
3) Main should open the file given in the argument
    a. If the file cannot be opened, an error message should be displayed and the program should end.
    b. Otherwise, the program should continue.
4) There should be a vector of <item*> type where all the contents will be put in
5) Main should read in the file into the vector in this manner:
    a. The file is a csv (comma separated values), so you will be looking for ',' to distinguish the data.
    b. If the thing you are reading says "item", perform the following
        i. Allocate memory for a new object of item type.
        ii. Read in the name, price, and bar code (in that order) and set the corresponding members into the new object.
        iii. Push the object into the vector of <item*> type
    c. If the thing you are reading says "fruit" perform the following
        i. Allocate memory for a new object of fruit type
        ii. Read in the name, price, bar code, sell by month, sell by day, sell be year (in that order) and set the corresponding members into the new object.

        iii. Push the object into the vector of <item*> type

    d. If the thing you are reading says “dairy” perform the following

        i. Allocate memory for a new object of dairy type

        ii. Read in the name, price, expiration month, expiration day, expiration year, and storage temperature (in that order) and set the corresponding members into the new object.

        iii. Push the object into the vector of <item*> type.

6) Display the contents of the vector in an ordered fashion using the display method. Make sure that the display called corresponds to the correct class, not just the base class.

HINT: You will need to make two virtual methods. I am not telling you which ones here, but there are two.

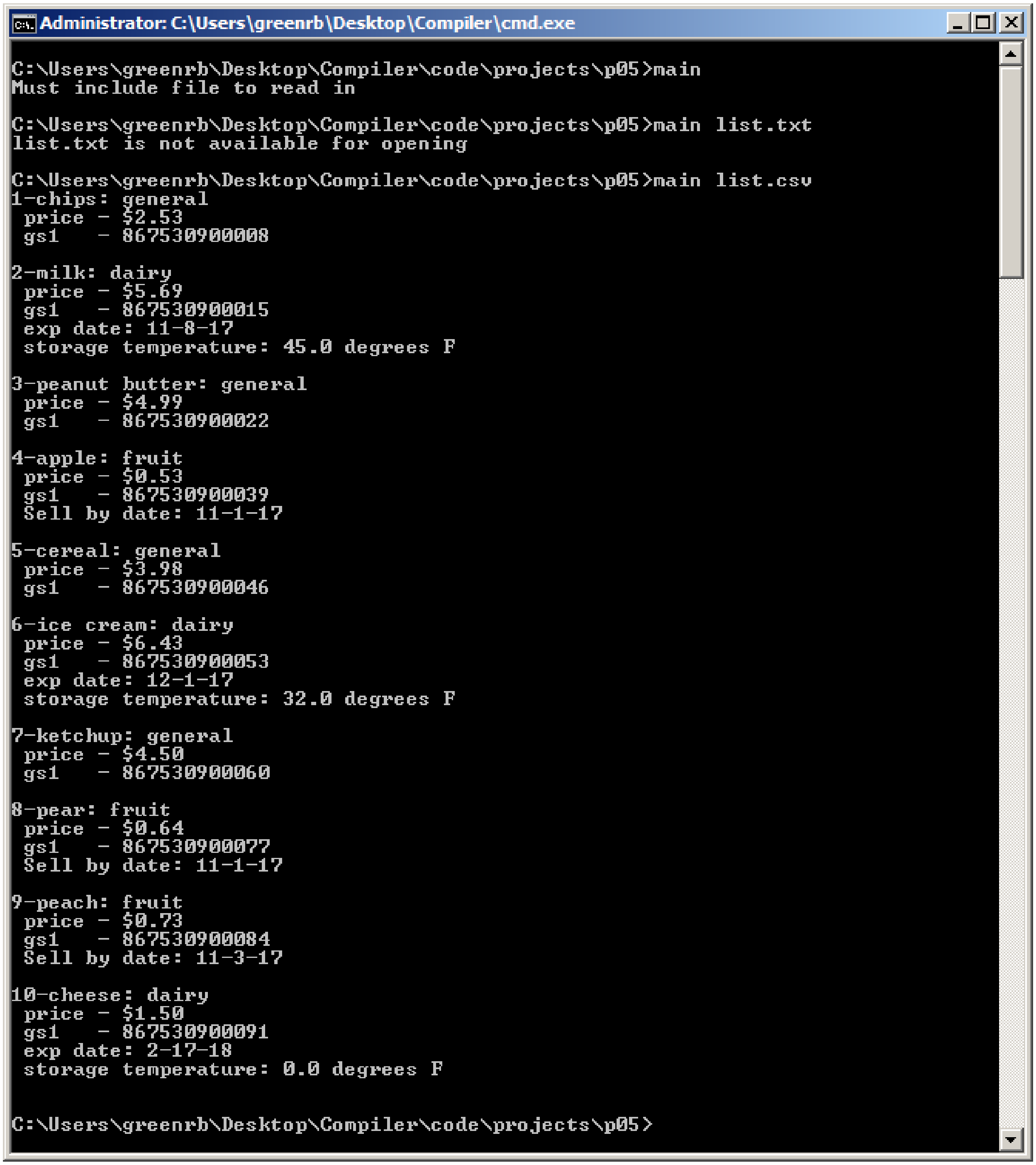
Grading:

Your submission will be graded using a couple of different .csv files that will not be given ahead of time. Please test your code with list.csv (provided) and some csv files of your own creation.

Submission:

Submit main.cpp, item.h, item.cpp, fruit.h, fruit.cpp, dairy.h, dairy.cpp, and a makefile that works for all .h and .cpp files.

Sample Output:



```
C:\Users\greenrb\Desktop\Compiler\code\projects\p05>main
Must include file to read in

C:\Users\greenrb\Desktop\Compiler\code\projects\p05>main list.txt
list.txt is not available for opening

C:\Users\greenrb\Desktop\Compiler\code\projects\p05>main list.csv
1-chips: general
 price - $2.53
 gs1   - 867530900008

2-milk: dairy
 price - $5.69
 gs1   - 867530900015
 exp date: 11-8-17
 storage temperature: 45.0 degrees F

3-peanut butter: general
 price - $4.99
 gs1   - 867530900022

4-apple: fruit
 price - $0.53
 gs1   - 867530900039
 Sell by date: 11-1-17

5-cereal: general
 price - $3.98
 gs1   - 867530900046

6-ice cream: dairy
 price - $6.43
 gs1   - 867530900053
 exp date: 12-1-17
 storage temperature: 32.0 degrees F

7-ketchup: general
 price - $4.50
 gs1   - 867530900060

8-pear: fruit
 price - $0.64
 gs1   - 867530900077
 Sell by date: 11-1-17

9-peach: fruit
 price - $0.73
 gs1   - 867530900084
 Sell by date: 11-3-17

10-cheese: dairy
 price - $1.50
 gs1   - 867530900091
 exp date: 2-17-18
 storage temperature: 0.0 degrees F


C:\Users\greenrb\Desktop\Compiler\code\projects\p05>
```