IIT Ropar
CSL201 Data Structures
Semester 1, AY 2018/19

Lab Assignment 1 - 40 marks Due on 28th August, 11:59 PM

Objective

To understand and use various advanced features of C++ programming language and OOP concepts.

Instructions

1. You are to use C++ programming language to complete the assignment.
2. Provide a Makefile to compile your final code.
3. All function and class declarations should be in ".h" files while the definitions should be in ".cpp" files. It is recommended that you have one header file and one cpp file for each class.
4. This is an individual assignment. You can have high level discussions with other students, though.
5. Include a "Readme.txt" file on how to compile and run the code.
6. Upload your submission to moodle by the due date and time.
7. Late submission not allowed.No assignment will be accepted through email after submission system closes. I suggest you to upload the assignment at least 1 hour in advance.
8. If you find assignment description vague, take appropriate assumption and write that in the Readme. Clearly state assumptions and tell those assumptions during viva.
9. If any student asks for deadline extension or sends assignment through email, he or she will get 5% penalty.

Program Description

In this assignment you need to implement a system for grocery store management.The grocery store has a number of items. Each item has at least the following attributes: purchase ID (long), location (x,y), and purchase time (take it as the time when you add an item to the system). Purchase ID is assigned by the program to each new item.

You have to do the following:

1. Write a class for grocery store management system. The class should have capability to add new items to the system. Maintain the items in a linked list. Do not use STL, implement your own linked list. Write a main function to implement overall system.
2. Implement a base class for items. The base class should have the common attributes as private variables, a constructor to initially add the item, and some public functions. Note that the public functions are called the interface of a class. At least two public functions should be there:
   print();// To print the details
   update();// To update the variable values.
   An item is the “element” of your linked list node. Obviously, now you need to define a node for linked list. You may want to use “friend class” and “virtual function” features here.
3. Define a new class for fruits, which extends the item class. Fruits have the following attributes: purchase ID, location, purchase time, type, weight per unit, calories, and price per unit. You may also need to add more functions to this class.
4. Extend the item class for vegetables. Vegetables have the following attributes: purchase ID, type, location, purchase time, price per kg, volume per kg. You may also need to add more functions to this class.
5. Read the fruit details from the file “fruits.txt” and add to the system using the add function of the class. The data is given in the following format:
   (type, location, weight per unit, price per unit, calories, units to add)
   apple, 3,4, 0.3, 15, 56, 100
6. Read the fruit details from the file “vegetable.txt” and add to the system using the add function of the class. The data is given in the following format:
   (type, location, price per kg, volume per kg, calories, amount to add (kg))
   onion, 3,4, 40, 30, 80, 5

7. Define an item node for the linked list of the main grocery store class. This node will mostly be similar to the one given in the book. An item is the “element” of the node class. If possible, maintain a single linked list for all items.
8. The main grocery store class should have a number of variables and functions, some private and some public. By default, make everything privacy. When you really have to, then convert that to public. Variables should be private only. One of the key variables is purchase ID that create every time you add a new item to the system.
9. Two main functions will be add item and remove item. When adding an item, you should put it into a node and add to the linked list. Similarly, while removing (selling) an item, if the quantity/amount is zero, you should remove the node and free the allocated memory, if any.
10. Add the functionalities given below the the main grocery class as public functions or interface. Although it may be inefficient, but implement these functionalities on the linked list.

Functionalities

The developed system should implement the following functionalities.

1. The program should be able to add fruits and vegetables to the system dynamically. User will enter the type of fruit/vegetable followed by its details.
2. The program should be able to sell fruits and vegetables. The user will enter type and count/amount. The program should tell how much to pay and update the amount/count in the system after selling. You should first sell the items that are close that are cheapest (assuming same type items may have different prices).
3. Given a fruit or vegetable as input type, the program should be able to display the current amount/count in the store.
4. The program should be able to display all items in the store with their details.
5. GIven an area, the program should be able to find the items in that area and display their details.
6. Given calories, find maximum amount/count of the vegetable/fruit and sell. I.e., the user may want maximum amount/count for given calories.

No need to implement a separate GUI. You can display these choices in terminal itself and let the user make a choice. The user can be asked to enter subsequent details depending on the choice.

Overall, the students should use the following C++ features at least once in the code with proper justification:

- Function overloading

- Operator overloading
- Constructors and destructors
- Inheritance
- Virtual functions
- Dynamic memory allocation
- Function templates
- Access specifiers

You are allowed to add new functionalities to the system to use these features. Please mention new additions during the assignment demo.

Assessment

- The marks division is as follows: 15 marks for features, 15 marks for functionalities, 10 marks for code organization and clarity.
- There will be assignment demo and viva. **You may be asked to change some functionality during the viva.** You should be able to do that if you wrote the code yourself.
- You will have to run the code directly from your Moodle submission, you won't be allowed to bring an updated version to the lab. ALthough the data files may change.
- After you run the code, we will ask you questions about the code. Be prepared to answer the questions.