CS200 Introduction to Programming
Spring 2022
Programming Assignment 3

Guidelines

Rules:

1. The objective is not simply to get the job done, but to get it done in the way that is asked for.
2. Any cheating case will be reported to Disciplinary Committee without any delay.

Coding Conventions:

- Constants are “ALLCAPS” or “ALL_CAPS”.
- Variables are “allsmall” or “all_small”.
- All function names must be “firstWordSmallAllOtherWordsCamelCase”.
- All class names must be “CamelCaseWords”.
- All curly brackets defining a block must be vertically aligned.
- Use tabs where needed instead of spaces.
- File naming: RollNumber_PA3.zip

Course Learning Objective (CLOs):

1. CLO1: Enabling Knowledge:
    a. (C1) use object-oriented programming model: abstract data types, encapsulation, inheritance, and polymorphism to code algorithmic solutions using standard coding conventions.
    b. (C1) use of fundamental features of an object-oriented language like C++.
2. CLO2: Critical Thinking and Analysis:
    a. (C4) analyze the requirements for solving simple algorithmic problems.
3. CLO3: Problem Solving:
    a. (C6) design algorithm and implement program code in an object-oriented programming language such as C++ to solve simple algorithmic computing problems, based on the analysis of the requirements.
    b. (C5) evaluate the correctness of the proposed solution.
4. CLO4: Communication:
    a. (C2) explain key concepts of algorithmic design in written form.
5. CLO5: Responsibility:
    a. (C3) apply relevant conventions, standards, and ethical considerations to writing computer programs.

Total Marks for the assignment: 100

Q1: **Total Marks = [40]**

You have to implement the system for PDC.

Consider the PDC Menu System. PDC maintains lists, each list stores information on what will be served for breakfast, lunch, and dinner.

You will have to implement a singly linked list to do this question.

The linked list will have a Node struct which has:

- Variables:
  - next : pointer
  - dish : string
  - quantity : int
  - price_per_dish : int
- Member functions
  - constructors – default and parametrized [3]

The LinkedList class will have functions that support the following operations:

1. Constructors (default, parametrized and copy), destructor [5]
2. Insert new dish to the lists [5]
3. Delete dish [5]
4. Update quantity [5]
5. Sum of all dishes [5]
6. Quantity of dish [2]
7. Update price [5]
8. Display Menu [5]

New dishes will be inserted through the tail of the list.

To “update quantity”, the list will look for that dish name and update its *quantity* variable. It will then return true. If the dish does not exist, the function will return false.

The “sum of all dishes” function will return the sum of all dishes’ quantity from the list or 0 if the list is empty.

“Delete dish” will search for a dish and delete it from the list.

The “quantity of dish” function will take a dish name as input and return its quantity. It will return -1 if the dish does not exist.

“Update price” function will take a dish name and an int value as input. It will then update the price of that dish, if it exists.

Display menu will show the dishes available and their prices. If a dish is sold out, that dish would not be displayed on the menu.

Question 2: **Total Marks = [60]**

This question builds upon question 1. Please make sure your previous question is functioning correctly.

For this question, we will analyze PDC sales and set up a system for PDC admin. For this we first need to simulate PDC sales.

We will create two menus: one for the customer side and the other for the PDC admin side.

Once the program starts running, ask the user what they would prefer to view: the customer view or the admin view. They can also quit. Display the appropriate information.

Quitting here will exit the program.

If the user choses customer view, display the menu of that day. The menu will then ask if the user wants to place an order or exit. If the user places an order, ask the dish name and its quantity.

However, if the user wants to quit, do not exit the program but take the user back to the start and ask if they want customer or PDC admin side view.

Create a list object, named *Orders*. *Orders* will keep track of all the orders that are sold. This list will keep information where each Node represents a single order. Remember a person can order more than one dish thus one person may have more than one order.

If the user choses to view the PDC admin side view:

The PDC admin side menu will support the following functions:

1. total sales	[5]
2. total number of orders per dish	[5]
3. total cash received per dish	[5]
4. most favorite (ordered) dish	[5]
5. least favorite (ordered) dish	[5]
6. Quit	[3]

In order to implement the above functionalities, you have to update the LinkedList class you created in question 1 with new functionalities.

Total sales will return the total cash received.

For total number of orders per dish, the function will take a dish as input and return how many times it has been ordered.

Total Cash received per dish will take as input a dish name and return how much cash has been received by selling that dish. Hint – use a maximum function on number of orders.

Most favorite dish will return the most ordered dish. Similarly, least favorite dish will return the least ordered dish. Hint – use a minimum function on number of orders.

Quitting will take you back to the first menu and the user will have to choose between customer and PDC admin view.

Implement these functionalities in the LinkedList class as you see fit.

You may implement helper functions or new class members where needed.

Marks Distribution:

- Menu display with error handling [10]
- Customer Side Implementation [5]
- Error handling and Corner Cases [5]
- Program does not crash unexpectedly and only exits when user tells it to [2]

Whole Assignment

- Comments and coding conventions followed [5]
- Correct program flow, correct error messages [5]

Submission Guidelines:

You will write your code in one .cpp file. Rename your file as rollNumber_PA2 and submit the zipped file.