CS 161B: Programming and Problem Solving II

Assignment 1: Food Cart Ordering

Grade Level A

Academic Integrity

You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the Student Code of Conduct at PCC.

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

Introduction

Have you ever felt the need to do some quick calculations when you order food? Let's write a menu-driven program that will do a few calculations based on the user's choice.

Purpose

In this assignment, you will be writing a menu-driven program to calculate how much a customer will pay once they are done ordering food at a food cart. You will give the user a menu with some choices, and let them pick a choice. Based on the choice they pick, you will ask them some questions and give them results. This process will repeat until they choose to quit the program. The purpose of this assignment is to modularize your program.

After completing this assignment you will be able to:

- Write functions to modularize your program and reduce redundancy
- Write functions that take parameters by value and return values
- Write void functions that take nothing and return nothing
- Use a do-while loop to display a menu to the user
- Use the correct exit condition to exit out of the loop, no breaks or returns inside the loop.

Task - This is the Grade A version.

What grade would you like?

We understand that your time and money is valuable, and we want to ensure that this programming assignment is tailored to your needs and circumstances. Whether you are a student juggling multiple commitments, a student dealing with personal responsibilities, or a student facing time constraints, or other issues in life, this assignment aims to provide programming skills at different levels that will meet the objectives of this week. With that in mind this assignment has been broken down into 3 different grade levels - Grades A (Exceeds), B (Meets) and C (Approaching). ***This is the Grade A version.***

Tasklist for getting a grade A:

The below set of tasks, if completed correctly per the rubric will get you a grade of **Exceeds** and help master the concepts required to learn the objectives for this week's topic. You can improve your grade by resubmitting the assignment once.

- ❏ Before you get started:
    - ❏ Check out Sample Assignment A01 - Algorithmic Design document
    - ❏ Check out Sample Assignment A01 - SampleA01.cpp
- ❏ Your job is to create a program that will help someone calculate how much a customer will pay once they are done ordering. They may order one or more items.
- ❏ You will need to obtain some information from the user:
    - ❏ Each item they want to order. This includes the cost of the item (using a floating point value) and optionally include a name (using a string)
    - ❏ If they have another item to their order. If yes, then get that information and add to the total cost. If "no", then go to the next step.

  - ❑ They may want to add a tip as well. Prompt them to enter the tip amount and add this to the final total.
  - ❑ Some discounts may apply for larger orders. If the total (including the tip) is greater than $50, they get a 10% discount, and if the total is greater than $35 and less than $50, they get a 5% discount.
- ❑ You must have functions to do the following:
- ❑ **`void welcome()`** - This function prints a welcome message to the user.
- ❑ **`void displayMenu()`**- This function prints the menu to the user. Your menu must have 2 items - Place an order and quit. Here is an example:
  1. Place an order
  2. Quit
- ❑ **`void readOption(int &option)`**.
  1. This function will read an option and return to the called function through the reference parameter.
  2. Call the **`readInt`** function to do this. You must catch all invalid data such as characters, negative numbers etc.
  3. You must also do validation to make sure that the number is 1 or 2 and nothing other than that. Use a while loop to do this.
- ❑ **`void readInt(string prompt, int &num)`** ***(****new for this level****)***
  1. This function should be used any time you read any integers from the user. Use this function to read the menu option from the user.
  2. It takes a string prompt, outputs it, reads a number from the user, validates and returns the num by reference.
  3. See Samplea01.cpp for the function. You may use the function in the sample.
  4. You must catch all invalid data such as characters, negative numbers etc.
- ❑ **`void readDouble(string prompt, double &num)`** ***(****new for this level****)***
  1. This function should be used any time you read any floats or doubles from the user. Use this function to read the cost of the item, tip, etc from the user.
  2. It takes a string prompt, outputs it, reads a number from the user, validates and returns the num by reference.
  3. Write it exactly like the **`readInt`** but declare a double or float instead of an int.
  4. You must catch all invalid data such as characters, negative numbers etc.
- ❑ **`void placeOrder(double &cost)`**
  1. Call this function if the user selects 1 from the menu.
  2. This function will go through a loop and ask the user to enter item names and their cost until the user answers ‘n’ to the question - `Do you want another item? (y/n):` You must validate to make sure the user enters y/n for this question (it should not be case sensitive - must accept both uppercase and lowercase). Use a while loop to do this.

3. The total cost must be returned through the reference parameter and be printed in main().
4. Call the `readDouble` function to do this. You must catch all invalid data such as characters, negative numbers etc.
5. Do not print anything in this function. The tip, discount, and final total must be printed in `main()`.

- ❏ `double tipDiscount(double &tip, double &discount, double cost).`
  1. This function takes the `cost` by value, and the `tip` and `discount` by reference.
  2. **The function prompts the user to enter a tip amount, and reads into the tip parameter.** This will be used to calculate the total, and it will also go back to `main()` so it can be printed in `main()`.
  3. Call the `readDouble` function to do this. You must catch all invalid data such as characters, negative numbers etc.
  4. It then calculates the discount based on the total (including the tip), and returns the final total to main(). The `discount` amount goes back by reference to be printed in `main().` See sample run.
  5. Do not print anything inside the function.
- ❏ `int main()`
  1. Declare all variables needed.
  2. Call the `displayMenu` function.
  3. Call the `readOption` function.
  4. There must be a do-while loop in main that will keep repeating the menu until the user enters 2.
  5. When the user enters 1, call the `placeOrder` function.
  6. Then call the `tipDiscount` function that will calculate the final total and return the value. It will also receive the tip and discount amount by reference.
  7. Print the tip, discount and final total when it comes back. Repeat the menu and continue until the user enters option 2.
  8. Print dollar amounts with a $ sign and output to 2 decimal places.
  9. When the user enters 2, quit the program after printing a thank you message.
- ❏ Open the Algorithmic Design Document, make a copy, and follow the steps to create your algorithm.
- ❏ You must express your algorithm as **pseudocode.**
- ❏ **Check out all the coding constructs under Criteria for Success.**

Criteria for Success

- ❏ Test your program using the following sample runs, making sure you get the same output when using the given inputs (in `blue`):

```
Welcome to my Food Cart Program!
What would you like to do today?
Pick an option from below:
        1. Place an order
        2. Quit
>> 9
Invalid Option! Please choose 1-2!
>> 1
Enter the name of your item: Pasta
Enter the cost of your item $: 15.75
Do you want another item? (y/n): y
Enter the name of your item: Bowl
Enter the cost of your item $: 12.75
Do you want another item? (y/n): y
Enter the name of your item: Soda
Enter the cost of your item $: 3.50
Do you want another item? (y/n): x
Invalid Option! Please choose y/n!
>> n

Your total is: $32.00

Enter the amount of tip you want to add $: 3.50

Your total is: $35.50
You get a 5% discount!
Your discount is $1.78
Your final total is: $33.73

What would you like to do today?
Pick an option from below:
        1. Place an order
        2. Quit

>> 1
Enter the name of your item: Fajita Bowl
Enter the cost of your item $: 20.75
Do you want another item? (y/n): y
Enter the name of your item: Vietnamese plate
Enter the cost of your item $: 22.75
Do you want another item? (y/n): y
Enter the name of your item: Soda
Enter the cost of your item $: 3.50
Do you want another item? (y/n): x
Invalid Option! Please choose y/n!
>> n

Your total is: $47.00

Enter the amount of tip you want to add $: 10.00
```

```
Your total is: $57.00
You get a 10% discount!
Your discount is $5.70
Your final total is: $51.30


What would you like to do today?
Pick an option from below:
        1. Place an order
        2. Quit

>> 2

Thank you for using my program!
```

- ❏ Check out Sample Assignment A01 - Algorithmic Design document
- ❏ Check out Sample Assignment A01 - SampleA01.cpp
- ❏ Complete zyLabs Chapter **9. CS161B: Functions Pass by reference**.
- ❏ **Follow these Coding Construct Requirements:**
    - ❏ Your program must have function prototypes. Place the prototypes for your functions globally, after your #includes, just before main(). All functions must be implemented after main().
    - ❏ Each function must do only what it is supposed to do. For example, the menu function will simply display the menu, the readOption will take the option from the user, and so on.
    - ❏ **You may not use a while true loop and break statements in the loop.**
    - ❏ **Your program must do all data validation mentioned in the functions.**
    - ❏ Try not to have any redundant code (repeated code) in your program. That is the purpose of functions.
    - ❏ When you display the numbers for cost, tip, and discount, it should be formatted appropriately with two decimal places.
    - ❏ Use constants where appropriate (for discount percentages).
    - ❏ **Do not use arrays or any vectors for this program. Use only the concepts and functions we have learned so far.**
- ❏ Please open and compare your work with the grading rubric before submitting.
- ❏ Remember to follow all style guidelines.
- ❏ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `a01.pdf`, and upload it to the D2L assignment by **Wednesday**.
- ❏ Upload your `a01.cpp` C++ source file to the D2L assignment by **Sunday**.
- ❏ Do your own work. Consult the syllabus for more information about academic integrity.

Additional Support

❏ Post a question for the instructor in the Ask Questions! area of the Course Lobby.