Programming Assignment 1

Filename: `pa1_surname.c`

In Florida, there is a simple café that serves only two items on its menu: coffee and pastry. All prices are set in whole dollars with no cents. Customers order whole numbers of these items. The total bill is split equally among everyone in the party (i.e., the party size).

The café owner has asked you to write a program that can quickly generate a receipt. The program should ask the cashier to enter the price of a coffee, the number of coffees ordered, the price of a pastry, the number of pastries ordered, and finally the party size. With this information, the program will calculate the total cost before tax, apply Florida's 6% sales tax, determine the final bill with tax included, and show how much each person in the party should pay.

To ensure the program handles values correctly, store the tax amount, total with tax, and each person's share in variables of type `double` so that decimal values are preserved. When printing these results, format the output to show exactly two decimal places using `printf()`.

Important Note

It is fine if the result of multiplying the "each person pays" amount by the party size does not exactly equal the total with tax. Such small differences are normal and may arise either from rounding in decimal arithmetic or from the way floating-point numbers are represented in memory. These cases were discussed in the lecture. See **Sample Runs 2 and 3**.

For this assignment, you may assume that the user always enters valid values that match the description. In short, prices, quantities, and party size will be positive whole numbers (i.e., greater than 0). Three sample runs are provided below. Anything that is underlined represents user input.

Sample Run 1

```
==== RECEIPT ====
Coffee price: 3
Number of coffees: 2
Pastry price: 4
Number of pastries: 3
Party size: 3

Total before tax: $ 18
Florida tax (6%): $ 1.08
Total with tax: $ 19.08
For a party size of 3, each person pays: $ 6.36
=== THANK YOU ===
```

Sample Run 2

```
==== RECEIPT ====
Coffee price: 5
Number of coffees: 1
Pastry price: 2
Number of pastries: 2
Party size: 4

Total before tax: $ 9
Florida tax (6%): $ 0.54
Total with tax: $ 9.54
For a party size of 4, each person pays: $ 2.38
=== THANK YOU ===
```

Sample Run 3

```
==== RECEIPT ====
Coffee price: 4
Number of coffees: 2
Pastry price: 3
Number of pastries: 1
Party size: 3

Total before tax: $ 11
Florida tax (6%): $ 0.66
Total with tax: $ 11.66
For a party size of 3, each person pays: $ 3.89
=== THANK YOU ===
```

Important Note

Refer to the General Instructions and Submission Guidelines for more guidance.

Grading Rubric

Criteria	0	5	10
Compiling the Source Code on the Eustis Server *The solution developed incorporated only concepts covered in class (except those explicitly allowed in the instructions).*	The source code could not be compiled on the Eustis server. Therefore, the entire submission is not graded anymore and will be marked as 0/100. **OR** The solution included code/s OR function/s not covered in class. Therefore, the entire submission is not graded anymore and will be marked as 0/100.	*(Not Applicable)*	The source code can be compiled on the Eustis server. **AND** The solution included code/s AND function/s that were covered in class.
Passed Sample Test Cases *For each of the sample runs provided. Max of 10 per test case.*	There is no implementation at all.	The program logic is incorrect; however, some effort was exerted (including cases where the program does not produce any output). **OR** The program produced partially correct output (e.g., incorrect prompt used; wrong spelling; missing characters/symbols; missing/exceeding whitespaces; or using the wrong case for letters).	The program passed all the sample runs provided where the output **matched exactly** the expected output.
Passed Secret Test Cases *The number of secret test cases will not be disclosed. However, they will be weighed such that the overall sum is 50%. Each secret test case will be binarily graded as either correct or not. Max of 50 as the sum for all test cases.*	*(Binary and Weighted; Refer to Description)*	*(Binary and Weighted; Refer to Description)*	*(Binary and Weighted; Refer to Description)*
Code Readability *Refer to the style guide here.*	The code was not very readable **AND** lacked meaningful comments.	Either the source code was not very readable because of poor indentation **OR** there was a lack of meaningful comments throughout the source code.	The source code was readable because proper indentation was followed. Further, relevant and meaningful comments were incorporated throughout the source code.

General Instructions and Submission Guidelines

Important. Please read the following instructions and guidelines carefully and in full. They will only be presented here in the first homework. However, they apply to every homework submission in this course unless stated otherwise.

All programs must compile and run correctly on the Eustis server. If your source code does not compile on the Eustis server, your submission will receive a grade of `0/100`, even if it compiles and runs on your personal machine. No exceptions will be made to this policy. The purpose of using the Eustis server is to ensure a consistent environment for grading. Submitting code that compiles, even if incomplete, is better than submitting code that fails to compile.

You are required to submit your codes on Webcourses for grading. Eustis is only a testing environment and does not submit your work automatically. It is your responsibility to ensure that your final code is uploaded to Webcourses. If you are short on time, you may skip compiling on Eustis. However, keep in mind that if your code fails to compile during grading, you may receive no credit.

Your program must read input from the standard input (keyboard), unless otherwise stated. In addition, you are required to include the following comment block immediately after the preprocessor directive in your source code. Replace the fields enclosed in < > with the appropriate information. Submissions without this comment block may not be graded.

Required Comment for Any Code Submission

```
/*
    COP3223C Fall 2025
    Programming Assignment <number>
    Student Name: <your complete name>
    File Name: <the required filename>
    NOTE: I hereby certify that this submission is my original work.
    It was completed independently by me without unauthorized assistance.
    I affirm that all sources consulted have been properly acknowledged.
    No part of this work was copied or plagiarized from any other source/s.
*/
```

Ensure that your source code is readable by following proper indentation. Include meaningful comments throughout your program to explain its logic. Remember, code readability and comments are part of the grading rubric. Therefore, missing or poor readability and comments will result in point deductions. Refer to the following document for the guidelines.

For each homework, you will submit exactly one source file, unless otherwise stated. The required filename will be clearly specified in the homework instructions (e.g., `pa1_yoursurname.c`). You must follow the naming convention exactly as described: Use your surname in lowercase. Do not include spaces, dashes, or special characters. Submissions with incorrect filenames may not be graded.

Note that Webcourses automatically adds a version number to the filename each time you submit. This is not a problem. Your most recent submission will be the one graded. You can submit multiple times until the due date. **Do not forget to click on the Submit button.**

Academic Integrity Policy

This is an individual activity. You must complete the assignment on your own. Collaboration of any kind is not allowed. You are not allowed to use generative AI tools (e.g., ChatGPT, GitHub Copilot) or receive code from other people. However, you are encouraged to consult other resources, such as textbooks, documentation, or class notes to help you understand concepts and design your solution. The implementation itself must be entirely your own work. Finally, when writing your solution, you are restricted to using only the codes or functions covered in class, unless otherwise stated. Using those that were not covered will result in a score of 0 for the entire submission.

Any submission suspected of academic misconduct will be immediately flagged and referred for formal investigation under university policy. Refer to the syllabus for the expectations and penalties.

If you have any questions or need clarification regarding the homework instructions, post them in the corresponding Discussion Forum on Webcourses. The instructional team will actively monitor and answer questions there.

Implementation and Grading Guidelines

For each homework, a set of sample runs will be provided. Each sample run represents a single and independent execution of your program. Without any modifications to your source code, your program must be able to handle all of these runs successfully. The output of your program is required to **exactly match the expected output** in every detail. This includes whitespace, capitalization, spelling, punctuation, and symbols. In the sample runs, anything that is underlined represents user input.

A detailed grading rubric will always be provided to show how each submission will be evaluated. Your submission will be evaluated by directly comparing its output to the provided sample input and output. In addition to these visible tests, your program will also be subjected to a separate set of **secret test cases** that are not disclosed in advance. These are designed to thoroughly evaluate the correctness, design, and logical structure of your solution. It is therefore your responsibility to design and test your program carefully to account for all possible scenarios that may arise from the problem description. Limiting your testing only to the provided examples is insufficient and will not necessarily earn you full credits. You are expected to think critically, test broadly, and ensure the robustness of your code. If certain assumptions are required for a homework, they will be clearly specified in the instructions.