

Homework Assignment #2 — The Coffee Shop

Abstract

After years of hard work, you've decided to quit your job and follow your dream alternative career: running a local coffee shop. You purchase the cheapest point-of-sale (cash register) software available, with the plan to write your own programs to supplement the software. In this assignment, you will write a program that analyzes the sales data by reading from and writing to text files.

Outcomes

After successfully completing this assignment, you should be able to:

- Develop a complex program in C.
- Read and write from text files
- Split C code into multiple source and header files and compile using a make file.

The Input Files

The sales for each day are stored in a simple text file. The contents of an example file are shown below.

```
0  523.21
3   2.50
10  3.50
3   2.50
5   1.99
```

The first column stores the item code for each sales item (0: the initial balance for the day, 3: ice coffee, 5: hot coffee, etc), and the second column stores the cost of the item.

The Assignment:

Write a program called `coffee_shop.c` that asks which task to perform using the following prompts:

Which program would you like to run: (1) Calculate overall revenue, or (2) Calculate register balance.

Please enter an input file:

If the option is invalid, print the following error message and exit the program:

This option is not valid.

Starter code is provided in `coffee_shop.c`

In a separate source file (you can choose the file name), write a 2 functions that do the following:

1. Read the input file and calculate the overall revenue for the day. Returns a float.
`float overall_revenue(FILE *f1);`
Parameter: A pointer to the open input file
Returns: the overall revenue for the day
2. Reads the input file and calculates the balance in the cash register at the end of the day.
`float balance(FILE *f1);`
Parameter: A pointer to the open input file
Returns: The balance in the cash register at the end of the day

Each function should be called from `main()`. After returning to `main()`, the revenue or balance should be printed to the terminal.

The function prototypes given above are suggestions. There are many ways to approach this assignment. You may add additional functions to your program.

Note: I am passing the pointer to the open file to each function, not the file name.

Example runs and outputs are found on Canvas.

Before Starting

Review the following topics:

- Input and output using `fprintf` and `fscanf` in C
- `makefiles`

Step 1: Read all instructions

Step 2: Prep work

Begin by writing pseudocode that outlines the algorithm you will develop. In addition to submitting your working program, you **must** submit pseudocode outlining your algorithm.

Step 3: Write your code

Using the pseudocode you developed in step 2, begin writing your algorithm in stages.

1. Write function stubs for each function that return 0.
2. Write `main()`. Best programming practice is to write and test outer loops and conditional statements, before working on inner statements. This allows you to catch bugs early, debugging as you write, instead of dealing with all the errors after you have many lines of code written.
3. Write each function, compiling and testing as you go.

Using your resources:

- If you consult web resources, or other students or staff when developing your program, *you must cite your source*. If you completed the entire program on your own, you should say in your write up file that this is entirely your work.
- If you research some or all of an algorithm from someone else or from somewhere else, *do not copy it*. Write it out in your own words and your own coding style. Also, please explain enough about how the algorithm works that the graders can conclude that you understand it.

Deliverables:

1. Write a document called **README.txt** that summarizes your program, how to run it, and detailing any problems that you had. If you used any outside resources for this assignment, be sure to cite your sources *and* explain in detail how the code works.
2. Your pseudocode, submitted as a .txt file or a picture of hand-written pseudocode.
3. `Coffee_shop.c` plus 1 additional .c file and a .h file.
4. `makefile` which will be used to compile your program. The executable should be called `sales.exe`

Programs submitted after 5:00pm on due date will be late and subject to the **Late Homework Policy**.

Implementation Notes

- You may not use arrays.
- For this assignment, you **must** partition your program into multiple files.

Grading:

- Correct execution with autograder test cases	50 pts
- Use of multiple source files and a makefile	10 pts
- Satisfactory README file with all required parts	10 pts
- Substantial pseudocode	10 pts
- Header comments for each function	10 pts
- Proper indentation	10 pts

Deductions:

- | | |
|---------------------------|---|
| 1. Does not compile: | - 10 pts and returned to student to correct |
| 2. Compiles with warnings | - 5 pts |
| 3. Use of arrays | - 5 pts and return to student to correct |

Example input and output can be found on the next page and on Canvas under the Homework 2 folder.

Example input and output. **User input is highlighted.**

Option 1:

Which program would you like to run: (1) Calculate overall revenue or (2) Calculate register balance.

1

Please enter an input file:

input.txt

83.14

Option 2:

Which program would you like to run: (1) Calculate overall revenue or (2) Calculate register balance.

2

Please enter an input file:

input.txt

208.14