

SEN-102 Introduction to Programming
Pre-Class Assessment Assignment
Last update: 2024-08-19

Introduction

This assessment activity is intended to determine whether you already have the skills associated with the SEN-102 (Introduction to Programming) competency. If you successfully complete this activity and receive a score of 75 or better, you have the option to get immediate credit for SEN-102 with the corresponding score as your mastery level (90-100, Legend or A, 75-89, Master or B). If you get lower than 75, this indicates that you have not mastered the SEN-102 skills and you must take the competency.

Read the instructions below, then write a program that follows these instructions. To submit your assessment work, you must **send your program by email to Dr. Sally within 14 days of receiving the assessment instructions**. Be sure to include all source code and any other files necessary to run your program.

This is an individual assignment. Everything you submit must be your own work. You may not discuss the problem with anyone else although you can ask me questions. You may not show your code to anyone else. You may not use any sort of AI tools. Any evidence of copying or borrowing will be treated as a violation of the university's policies regarding academic integrity, as described in the Handbook.

I may give different students slightly different problems, to make it possible to detect copying. Please work on your own! This is your future.

Please note the following:

- You must write your program in ANSI standard C. Although the essential skills of programming can be learned in any language, the *SEN-107 Fundamental Data Structures and Algorithms* competency, which follows from SEN-102, requires basic proficiency in C programming.
- The prohibition against using AI tools includes using any tool to translate from Python or another language into C.
- Your program must follow the C coding standards used in SEN-102. This includes naming standards, commenting standards and formatting standards. If the code you submit does not follow these standards, I will suspect that you used some AI tools. You will find the C coding standards on the SEN-101 Canvas course, under *Useful Resources*: <https://cmkl.instructure.com/courses/524/modules>
- You may of course use online reference information such as <https://www.w3schools.com/c/index.php> You can find two free ebook textbooks in the SEN-101 Canvas course, under *Useful Resources*: <https://cmkl.instructure.com/courses/524/modules>

Detailed Instructions

Write a program in C that correctly solves the following problem. *Your program must run on Linux as well as on whatever operating system you use on your own computer.*

VendorInfo Program Specification

Write a program called ***vendorInfo.c*** to manage income and inventory for vendors in a market. Each vendor sells a different type of merchandise, at a different price.

Start by asking the user for the name of the input data file. The input data for this program is a text file with four data fields separated by spaces, in the following format:

```
[VendorName] [Merchandise] [InventoryCount] [ItemPriceBaht]
```

For example:

```
Mook earrings 30 79
Woot teeshirts 23 120
Akraya donuts 35 22
Champ phonecases 15 189
Pee jeans 9 350
```

Your program should handle a maximum of 20 vendors. If an input file contains more than this number of lines, you should ignore the excess lines. I will give you some sample files.

A correct input file should not include any duplicate names or duplicate merchandise. (See below for a discussion of error checking.)

After reading and storing the vendor information, the program should repeatedly ask the user what they want to buy, and how many of these items. Your program must print out the vendor who sells this type of merchandise and the total amount owed for this purchase. The program must also keep track of the vendor's inventory (the number of items available). Every purchase will decrease the inventory.

All string comparisons in this program should be ***case-insensitive***. It should not matter if the user types "Luggage", "luggage", "LUGGAGE", etc. If there is a vendor who sells this item, you should find a match.

If the user tries to buy a type of merchandise but the vendor does not have enough items to satisfy the order, the program should print out a message explaining this, and not complete the sale.

The buying cycle should repeat until the user types "DONE". ***The cycle should also end if all the vendors are completely out of merchandise.***

Before the program exits, it should print out a list with each vendor's name, the total income for that vendor during the program run, and the number of items remaining in their inventory.

About Error Handling

Your program can assume that the data in the input file follows the correct format, with two exceptions:

1. You must check for duplicate merchandise.
2. You must check for a duplicate vendor name.

If you find either of these error cases, print a message and skip that line in the file. Note that the comparisons must be case-insensitive. That is, “Mook” and “mook” should be treated as duplicates.

The program must handle the following user input errors:

- Request for a type of merchandise that is not sold by any of the vendors;
- Entering a number of items that is not an integer.

In both these cases, print an informative message and ask again.

The program must also behave correctly if the user supplies the name of a data file that does not exist.

You will find a sample run of this program below. *Italic red text* is user input. This run uses the sample input file `vendor1.data`. Remember that your program must work with **any** input file that follows the specifications. Remember also that if there are more than twenty vendors in a file, the excess vendors should be ignored.

```
Enter name of input file: vendor1.data
```

```
>>> Read 12 vendors from file vendor1.data
Jay sells shoes for 300 baht each           Inventory 22 items
Mark sells dvds for 200 baht each          Inventory 10 items
Mook sells earrings for 79 baht each        Inventory 30 items
Woot sells teeshirts for 120 baht each      Inventory 23 items
Akraya sells donuts for 22 baht each        Inventory 35 items
Champ sells phonecases for 189 baht each    Inventory 15 items
Pee sells jeans for 350 baht each           Inventory 9 items
Chaiwat sells pencils for 12 baht each      Inventory 45 items
Namhom sells perfume for 225 baht each      Inventory 13 items
Ae sells socks for 22 baht each             Inventory 18 items
Q sells belts for 139 baht each             Inventory 10 items
Bright sells books for 120 baht each        Inventory 30 items
```

```
What do you want to buy? (DONE to end) jeans
How many do you want to buy? 9
You owe Pee a total of 3150 baht
What do you want to buy? (DONE to end) jeans
How many do you want to buy? 10
Sorry -- Pee does not have 10 items available
What do you want to buy? (DONE to end) tomyamkung
No vendor in this market sells tomyamkung
What do you want to buy? (DONE to end) phoneCASES
How many do you want to buy? 3
You owe Champ a total of 567 baht
What do you want to buy? (DONE to end) shoes
How many do you want to buy? Abc
You must enter a positive whole number for the number of items
What do you want to buy? (DONE to end) shoes
How many do you want to buy? 3
You owe Jay a total of 900 baht
What do you want to buy? (DONE to end) DONE
```

```

>>>End of day totals
Jay made 900 baht today      Remaining inventory 19 items
Mark made 0 baht today       Remaining inventory 10 items
Mook made 0 baht today       Remaining inventory 30 items
Woot made 0 baht today       Remaining inventory 23 items
Akraya made 0 baht today     Remaining inventory 35 items
Champ made 567 baht today    Remaining inventory 12 items
Pee made 3150 baht today     Remaining inventory 0 items
Chaiwat made 0 baht today    Remaining inventory 45 items
Namhom made 0 baht today     Remaining inventory 13 items
Ae made 0 baht today         Remaining inventory 18 items
Q made 0 baht today          Remaining inventory 10 items
Bright made 0 baht today     Remaining inventory 30 items

```

Evaluation of Your Work

This assignment is designed to assess most of the skills in SEN-102 including conditionals, iteration, simple data structures, using functions and file input/output. My evaluation of your submission, which will be on a scale of 0 to 100, will reflect your use of all these skills. To receive a top score, your program must:

- Follow all instructions and implement all required functionality;
- Produce correct results under all conditions stated in the instructions;
- Handle the error conditions cited in the instructions;
- Use functions to make the program more readable and understandable;
- Follow the coding standards, to use clear code formatting and comments; specifically you should have a comment for each function, indicating its purpose, input and output, and you must have a comment at the start of the program, with your name and the date.

I will test your program with different data files than the sample files I give you.