

Assignment 1: Shopping Calculator

Description

You will create a simple shopping calculator. Some items such as candy, eggs, and flour will have special promotional discounts applied to them. Users will enter the name of an item found at the grocery store, along with number of items purchased, and price per item. If the item name is found in a list of discounts, each additional item purchased will lead to the price being discounted by 5%, up to a maximum of 20%. See below for more details.

Both the item number and the item price entered by the user *must be numerical*, you can use a string function to check this. Once the user has finished entering items and their prices, your program will print a receipt. Each item will be printed along with item number and the price. Your program will print a total on a new line, and on another new line, the total discount applied to all purchases.

Discount

Here's the full list of foods that have the promotional discount. You should put these into a **list**:

- candy
- eggs
- flour
- hummus
- ice cream
- chicken soup
- diapers

Your script should compare the item name with the items in this list. If an item is eligible for a discount, each item the user purchases over 1 will cause the price for *all* of that item to be discounted by an additional 5%. For example:

- 1 item of candy costs 1.29. This is a discounted item but the user is purchasing only 1. They are not eligible for a discount.
- 2 items of candy cost 1.29, but since this is a discounted item a 5% discount is applied. Each item of candy now costs 1.2255, for a total cost of 2.451. The amount should appear as '\$2.45' on the receipt.
- 3 items of candy cost 1.29, but now a 10% discount is applied. Each item of candy now costs 1.161, for a total cost of 3.483. This will appear as '\$3.48' on the receipt.

The maximum possible discount should be 20%.

Formatting Currencies

Please note: you can correctly format Canadian currencies using f-strings. Your prices should have two decimal places at all times. For example:

```
price = 18.8923 # notice there are too many decimal places
print(f"{price : .02f}") # : .02f specifies float, 2 decimals with
zeroes to pad
```

Sample Output

Shopping Calculator

```
Please enter an item of food, or press Enter to exit: eggs
Item is: eggs. Please enter the price for this item: 3.49
Item is: eggs. How many will you purchase: 2
Please enter an item of food, or press Enter to exit: pork
Item is: pork. Please enter the price for this item: 7.09
Item is: pork. How many will you purchase: 1
Please enter an item of food, or press Enter to exit: candy
Item is: candy. Please enter the price for this item: 1
Item is: candy. How many will you purchase: 4
Please enter an item of food, or press Enter to exit: soup
Item is: soup. Please enter the price for this item: um
Error: price must be a number. Example: 1, or 1.99.
Item is: soup. Please enter the price for this item: 2
Item is: soup. How many will you purchase: 1
Please enter an item of food, or press Enter to exit:
```

RECEIPT

```
1. eggs      2 x $ 3.32 = $ 6.64
2. pork      1 x $ 7.09 = $ 7.09
3. candy     4 x $ 0.85 = $ 3.40
4. soup      1 x $ 2.00 = $ 2.00
Total:                      $ 19.13
You saved:                   $ 0.95
```

First Milestone

For your first milestone, you will need to submit the following:

Program Requirements

Program requirements are used as a planning tool to communicate with the client, to identify possible issues and to clarify program behaviour. The objective is not to 'set things in stone', but to create a plan for the next stage of development.

Write out in a Word file the requirements for this program. Use the example in lab 2 to help you do this. Your first milestone does not need to include

any Python code at this point. If there are program requirements that you are unclear about, you can also include questions about the functioning of the program here. Think about the **datatypes** that you will need in order for your program to function. What things need to be saved? Finally, think about what things can go wrong: what types of user input will cause issues, and what types of checks we will need to do to avoid errors. **You will be graded on how specific you can make your program requirements, and how much thought you've put into your questions.** Submit the file to Blackboard as a .docx file.

Second Milestone

For your second milestone, you will need to submit the following:

Flow Chart

Flow charts are a tool used for planning. Our field has two challenges: we need to write code that has the correct syntax for our programming language, and we need to break down program requirements into well-defined logic. Often when we are trying to accomplish both of these things simultaneously, we can get confused and overwhelmed. The objective of using a flow chart is to tackle the logical challenge first, and then to deal with correct syntax later.

Create a Flow Chart using an appropriate tool. Use the example in lab 3 to help you do this. You may use any application or tool you like to do this, as long as the flow chart follows the conventions taught in the lab. **You will be graded on how much thought you've put into solving the program requirements in your flow chart.** Again, in the second milestone it's not required that you provide code, only the various checks, loops and conditions that you will be using.

The Final Submission

Name your assignment `assign1.py`. This should be a working script that I can run on my computer, which will fulfill the program requirements. Here is how your program should look when I am running it:

Additionally, your submission should:

- contain a docstring at the top, with your name, myseneca ID, and description of the program. For example:

```
'''
Name: Eric
Myseneca: esmith1
Description: Enter your description of the program here.
'''
```

- Your script **must not** import any modules.

- Your script must contain **at least one function**.
- It's not necessary to justify the items and the prices, like I have done in the example. However, your prices should be **correctly displayed**, with two fixed decimal numbers for each price.
- For example: 2.00 instead of 2. Use [this guide](#) for help.
- Your script must **use the design patterns taught in class**. Code that obviously comes from external sources will be flagged for Academic Integrity violation.
- You may, however, use any of the built-in methods in Python. For example: `.lower()`, `.count()`, etc.
- Your script must contain documentation, including:
 - function-level docstrings
 - in-line comments explaining *why* you are doing what you're doing in the code.

Ideally, your script should run without errors. However if there are errors, I will award partial marks if you're on the right track.

You will submit `assign1.py` to Blackboard.

Academic Integrity

A reminder that the college has a **Zero Tolerance Policy** in regards to academic integrity. In the professional world, an employee who plagiarises is a huge liability for the company. Code falls under intellectual copyright laws: incidents involving stolen have caused litigation and millions of dollars in legal fees for companies. In the professional world, being caught stealing code **will** result in termination and possibly legal action.

On the other hand, it is expected that junior developers will make mistakes, hit roadblocks, get frustrated, get confused, introduce bugs and need help from a senior developer or mentor. Events are accepted and even celebrated as 'part of the process.' It's always better to ask for help: **knowing when you need help is just as important a skill as learning to code.**

Even if your code is not working or is incomplete, you will most likely get partial marks for what you have accomplished. Submitting code that is not your own, however, will always get a zero.

What Makes It Plagiarism?

You can absolutely communicate with classmates, and refer to outside sources. When you are discussing problems, use *pseudocode* or flow charts to explain. Sharing one or two lines of code is acceptable, sharing more than two lines of code is not. **Never share your assignment with other**

students, even if they say it is for reference. You will be charged with an academic integrity violation and receive a zero.

Rubric

Concept	Points
First Milestone	2.5
Second Milestone	2.5
Basic Function	5
Error Handling	2.5
Documentation and Comments	2.5
Total	15