

Assignment

Write a program to compute the sales for a coffee shop. The requirements are listed below:

- this coffee shop sells four types of coffee: Cappuccino, Espresso, Latte and Mocha
- a customer may pay extra money for options such as "large cup" and/or "cold drink" when buying coffee
- the user is allowed to enter the quantity and choose the preferred options by answering Y or N,
- display the current order summary including type of coffee, quantity, options requested and total amount when the user purchases a coffee that s/he wants.
- display sales statistics after user confirms his / her order.

A sample output is shown below:

```
===== RESTART: C:/Python38/assignment.py =====
Welcome to Coffee Shop System.

Coffee Shop Menu:
No. | Coffee Type | Price
0   | Cappuccino  | $25
1   | Espresso    | $20
2   | Latte       | $15
3   | Mocha       | $30
Please input your choice. Press "Enter" to confirm order (0 - 3):0
Please input quantity:2
Large Cup required? +$5.00 (Y / N):n
Cold required? +$3.00 (Y / N):N

Current Order Summary:
Cappuccino 2 cups: $50
Total: $50

Coffee Shop Menu:
No. | Coffee Type | Price
0   | Cappuccino  | $25
1   | Espresso    | $20
2   | Latte       | $15
3   | Mocha       | $30
Please input your choice. Press "Enter" to confirm order (0 - 3):

Statistics of Coffee Shop:
Total number sales = 1
Lowest Sales Amount = $50
Highest Sales Amount = $50
Total Sales Amount = $50
Average Sales Amount = $50.0
List of number of cups coffee sold:
    Cappuccino: 2

Coffee Shop Menu:
No. | Coffee Type | Price
0   | Cappuccino  | $25
```

User's input

Small letter or capital letter of y/n
should also be acceptable

User pressed Enter to confirm order here

```
1 | Espresso | $20
2 | Latte    | $15
3 | Mocha    | $30
Please input your choice. Press "Enter" to confirm order (0 - 3):1
Please input quantity:3
Large Cup required? +$5.00 (Y / N):y
Cold required? +$3.00 (Y / N):n
```

```
Current Order Summary:
Espresso 3 cups with option LARGE CUP: $75
Total: $75
```

```
Coffee Shop Menu:
No. | Coffee Type | Price
0   | Cappuccino  | $25
1   | Espresso    | $20
2   | Latte       | $15
3   | Mocha       | $30
Please input your choice. Press "Enter" to confirm order (0 - 3):2
Please input quantity:1
Large Cup required? +$5.00 (Y / N):n
Cold required? +$3.00 (Y / N):y
```

```
Current Order Summary:
Espresso 3 cups with option LARGE CUP: $75
Latte 1 cups with option COLD: $18
Total: $93
```

Two items are inputted by user

```
Coffee Shop Menu:
No. | Coffee Type | Price
0   | Cappuccino  | $25
1   | Espresso    | $20
2   | Latte       | $15
3   | Mocha       | $30
Please input your choice. Press "Enter" to confirm order (0 - 3):
```

```
Statistics of Coffee Shop:
Total number sales = 2
Lowest Sales Amount = $50
Highest Sales Amount = $93
Total Sales Amount = $143
Average Sales Amount = $71.5
List of number of cups coffee sold:
  Cappuccino: 2
  Espresso: 3
  Latte: 1
```

User pressed Enter to confirm order here

Total number of sales should
be 2 instead of 3 in this case

```
Coffee Shop Menu:
No. | Coffee Type | Price
0   | Cappuccino  | $25
1   | Espresso    | $20
```

```
2 | Latte | $15
3 | Mocha | $30
Please input your choice. Press "Enter" to confirm order (0 - 3):3
Please input quantity:3
Large Cup required? +$5.00 (Y / N):y
Cold required? +$3.00 (Y / N):y
```

Current Order Summary:

Mocha 3 cups with option LARGE CUP and COLD: \$114
Total: \$114

Coffee Shop Menu:

No.	Coffee Type	Price
0	Cappuccino	\$25
1	Espresso	\$20
2	Latte	\$15
3	Mocha	\$30

Please input your choice. Press "Enter" to confirm order (0 - 3):

Statistics of Coffee Shop:

Total number sales = 3
Lowest Sales Amount = \$50
Highest Sales Amount = \$114
Total Sales Amount = \$257
Average Sales Amount = \$85.66666666666667
List of number of cups coffee sold:
Cappuccino: 2
Espresso: 3
Latte: 1
Mocha: 3

User pressed Enter to confirm order here

Coffee Shop Menu:

No.	Coffee Type	Price
0	Cappuccino	\$25
1	Espresso	\$20
2	Latte	\$15
3	Mocha	\$30

Please input your choice. Press "Enter" to confirm order (0 - 3):
Current Sales Order is empty

User confirmed order without any items

Coffee Shop Menu:

No.	Coffee Type	Price
0	Cappuccino	\$25
1	Espresso	\$20
2	Latte	\$15
3	Mocha	\$30

Please input your choice. Press "Enter" to confirm order (0 - 3):

Following the steps below to implement the program:

1. Declare two constant variables named `LARGE_CUPE_PRICE` and `COLD_PRICE` to represent the price of each extra option (see the prices shown in the sample output above)
2. Declare a constant tuple named `"COFFEE_NAME_AND_PRICES"` to store the coffee name and prices. For each tuple, there is a separate tuple to store the coffee name and prices separately.
3. Declare the following constants to locate the coffee name and price from the constant tuple named `"COFFEE_NAME_AND_PRICES"` easily:
 `INDEX_COFFEE_NAME = 0`
 `INDEX_COFFEE_PRICE = 1`
4. Declare the following constants for index of tuple for each sales record.
 `INDEX_COFFEE_NO = 0`
 `INDEX_QUANTITY = 1`
 `INDEX_LARGE_CUP = 2`
 `INDEX_COLD = 3`
5. Declare the following variables inside the main function:
 - Declare a list `"current_sales_list"` to store sales items not yet confirmed by customer.
 - Declare a dictionary `"total_cups_of_coffee_sold"` to accumulate the quantity sold for each type of coffee. It is more convenient to use the coffee name as the key of this dictionary rather than coffees' ID
 - Declare the following instance variables:
 - `total_number_sales`: accumulate total number of sales
 - `highest_sales_amount`: the highest sales amount among all sales items; initialized to 0
 - `lowest_sales_amount`: the lowest sales amount among all sales items; initialized to `sys.maxsize`
 - `total_sales_amount`: accumulate total sales amount of all sales items.

The following diagrams show the relations between constant tuple, list and dictionary:

tuple **COFFEE_NAME_AND_PRICES**

0	0	"Cappuccino"
	1	25
1	0	"Espresso"
	1	20
2	0	"Latte"
	1	15
3	0	"Mocha"
	1	30

list **current_sales_list**

0	sales tuple 1	
	0	Coffee no (0-4)
	1	Quantity (Integer)
	2	Large cup (Y/N)
	3	Cold (Y/N)
1	sales tuple 2	
	0	Coffee no (0-4)
	1	Quantity (Integer)
	2	Large cup (Y/N)
	3	Cold (Y/N)
:	:	
n	sales tuple n	
	0	Coffee no (0-4)
	1	Quantity (Integer)
	2	Large cup (Y/N)
	3	Cold (Y/N)

dictionary **cups_of_coffee_sold**

"Cappuccino"	2
"Espresso"	3
"Latte"	1
"Mocha"	4

6. Write a function `compute_sales` which accepts four parameters (integer coffee no, integer quantity, string large cup Y/N, string cold Y/N). This function does the following:
- get the coffee type (by coffee no.), quantity, and any extra options of this coffee.
 - compute the amount of this coffee.
 - the coffee price can be obtained by the following statement:
`coffee_price = COFFEE_NAME_AND_PRICES[coffee_no][INDEX_COFFEE_PRICE]`
 where `coffee_no` should be an integer between 0 – 3, which is inputted by user.
7. The main function should contain the following logics:
- The program should display a welcome message and followed by a coffee shop menu.
 - To buy a coffee, a user will enter the coffee type, the quantity, and the options wanted (by entering Y or N). A current order summary would be displayed listing all the ordered items and the total amount of this order. User is allowed to place an order when there is items in this order item list.
 - To confirm order, a user will press "Enter" directly in the coffee shop menu without entering the coffee type. A statistics of coffee shop would be displayed including total number of sales, total, average, highest and lowest of sales. It will also display the quantity sold for each type of coffee.

Hints:

- a. For all yes no inputs (Y / N), it is suggested to convert all user inputs to small letters or capital letters (using `.upper()` or `.lower()` functions) before comparing using if condition.
- b. Giving `i` is a tuple in the `current_sales_list`, the coffee no, quantity, coffee options, coffee name and coffee price of a sales record can be obtained by the following statements:
- ```

coffee_no = i[INDEX_COFFEE_NO]
coffee_quantity = i[INDEX_QUANTITY]
large_cup = i[INDEX_LARGE_CUP]
cold = i[INDEX_COLD]
coffee_name = COFFEE_NAME_AND_PRICES[i[INDEX_COFFEE_NO]][INDEX_COFFEE_NAME]
coffee_price = COFFEE_NAME_AND_PRICES[i[INDEX_COFFEE_NO]][INDEX_COFFEE_PRICE]
```

**Instruction to students:**

1. This is an End of Module Assessment and the weighting of this assignment is 20% of the Module Mark.
2. This assignment should be done by each individual student. Plagiarism will be treated seriously. All assignments that have been found involved wholly or partly in plagiarism (no matter these assignments are from the original authors or from the plagiarists) will score Zero mark.
3. You must use Python 3 to develop the programs.
4. Your programs must follow the style guide stated in PEP8 – Style Guide for Python Code published by python.org. <https://www.python.org/dev/peps/pep-0008/>. Marks may be deducted if the style guide is not followed.
5. You are required to hand in:
  - Well-commented source code.
  - A test plan showing the evidence of testing.

| ID | Test Case Name                                                              | Procedure                                                                                                                                                         | Expected Output                                                                                                                                                                                                                                      | Result                 |
|----|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 1  | Calculate the total price for coffee Cappuccino with Large Cup Option       | 1. In the coffee shop menu, input '0' for coffee Cappuccino<br>2. Input '2' for quantity<br>3. Enter 'y' for large cup required<br>4. Enter 'N' for cold required | The correct output "Cappuccino 3 cups with option LARGE CUP: \$75" with total price of this order (i.e. \$ 75) should be displayed                                                                                                                   | Pass / <del>Fail</del> |
| 2  | Calculate the Statistics of Coffee Shop after inputting the test case ID 1. | 1. Press "Enter" in the coffee shop menu after inputting the test case ID 1                                                                                       | The correct output should be displayed:<br>Total number sales = 1<br>Lowest Sales Amount = \$50<br>Highest Sales Amount = \$50<br>Total Sales Amount = \$50<br>Average Sales Amount = \$50.0<br>List of number of cups coffee sold:<br>Cappuccino: 2 | Pass / <del>Fail</del> |
|    | ...                                                                         |                                                                                                                                                                   |                                                                                                                                                                                                                                                      |                        |
| 5  | Input invalid coffee number                                                 | 1. In the coffee shop menu, input '4'                                                                                                                             | An error message "Invalid coffee number. Please select again!" should be displayed and the coffee shop menu should be shown again.                                                                                                                   | Pass / <del>Fail</del> |

6. Prepare a word document with a number of test cases showing different inputs for different situations that your program may encounter and how your program responses to show the capability of your program.
7. For each test case, states the objective of the test case, input data and expected result. You should also include screen dump for each test run as evidence.
8. Submit all your works (in a zip file under the name of your student ID – e.g. 209999999.zip) to the Moodle website (<http://moodle2021.vtc.edu.hk>) by 11:55pm, 23 November 2020 (Monday). Late submission may score ZERO marks.
9. Each student will be required to conduct an assignment demonstration during laboratory class to show the system and walk through all the functions. The date of the demonstration will be after the assignment submission and is to be confirmed later (It should be the next lab session just after the submission of assignment normally). Marks will be deducted if a student does not attend the assignment demonstration.

---

10. Marks Distribution

- System Implementation (70%)
- Validation on the input data and display appropriate error messages (20%)
- Test plan (10%)