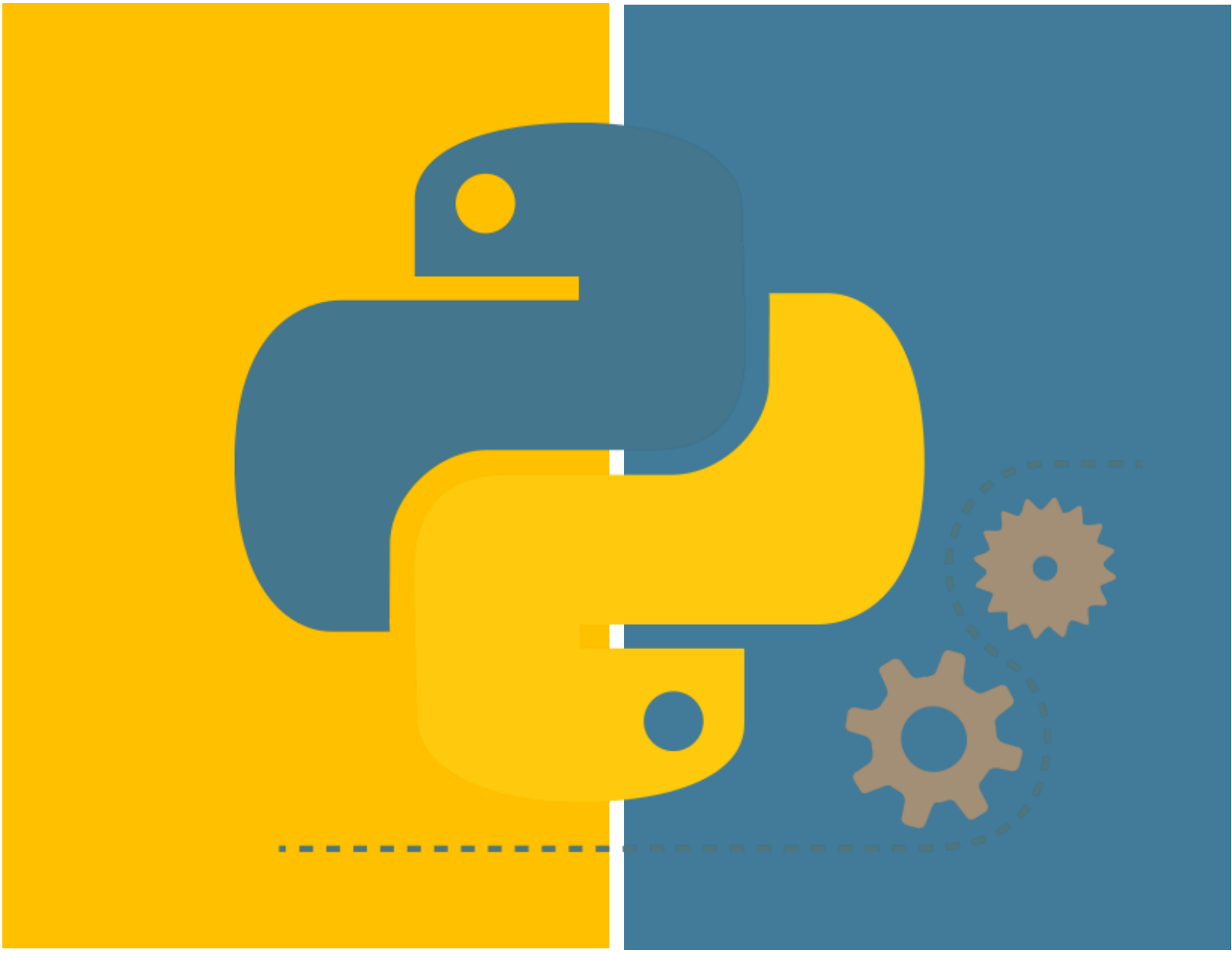
APRIL 26, 2022

SPDA | ASSIGNMENT 1

Object-Oriented Programming

STOCK MANAGEMENT IN FOOD STORE



Fatma EMIN -- 323713

PROBLEM

Question:

Write a program in terms of OOP to manage the stock in a Food Store.

Description:

A Food Store, selling the below mentioned items:

Item	**Quantity** (in numbers)	**Price** (per unit) (in PLN)
Bread	20	2
Muffins	50	3
Cake	30	2.5
Cookies	100	0.5

A program is required to show the following:

- ✓ Price list of items
- ✓ Initial stock availability
- ✓ Initial stock value of each item
- ✓ Purchase details
- ✓ Updated stock after purchase
- ✓ Updated stock value after purchase

Approach:

- Set the class as Foodstore.
- Define the set of items' quantity and price as dictionaries.
- Set the hierarchy as: items, quantity and price.
- Add the above list of items as initial stock – by calling a method to add the items' name, quantity and price.
- Display the initial stock and stock value after evaluation.
- User will be prompted to provide purchase details.
- User can enter the number of distinct items he is going to purchase, item name and quantity. Below validations are incorporated in the user inputs:
  - If the input number of purchase items is greater than 4, system will generate a ValueError – number of items cannot be greater than 4.
  - User can select items only from the above list of 4 items. If the input item name is different, system will generate ValueError – Order can be placed only for the above items.
- Purchase order will be displayed.

- System will update the stock by calling corresponding method. Below validations are incorporated:
    - If the ordered quantity is greater than stock balance of the ordered item, system will generate ValueError – Required quantity is not available.
- Updated stock and stock value will be displayed after purchase.

Class hierarchy:

Coding:

.py file attached.

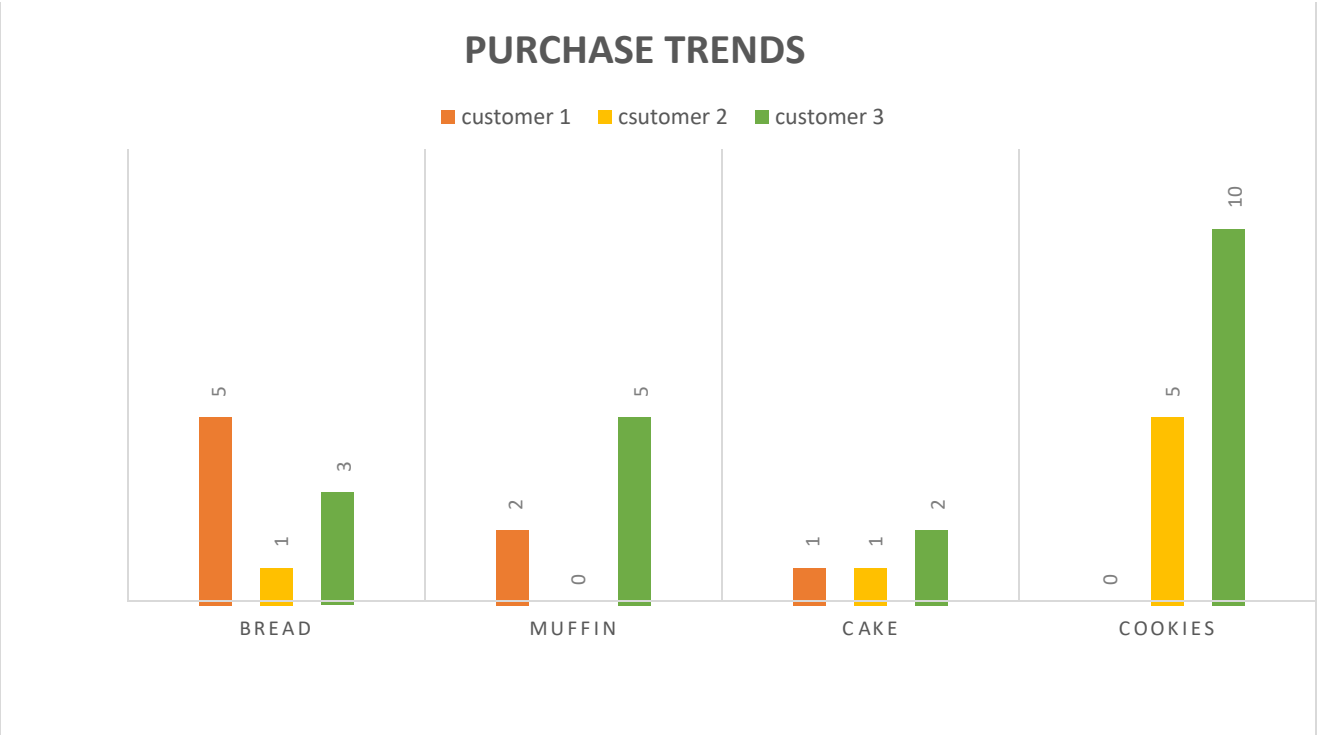
*Assignment done in Spyder.

Output:

```
....WELCOME.....


 Price List (in PLN)
----------------------
 Bread   - 2
 Muffins - 3
 Cake    -2.5
 Cookies - 0.5


 *** Stock Available (in numbers)***
-----------------------------------

{'Bread': 20, 'Muffins': 50, 'Cake': 30, 'Cookies': 100}

*** Stock Value of each item (in PLN) ***
-----------------------------------------

{'Bread': 40, 'Muffins': 150, 'Cake': 75.0, 'Cookies': 50.0}

How many distinct items you are purchasing?
1

Enter the item - Bread / Muffins / Cake / Cookies
Cake

Enter the quantity
5

>>>>>>>>> YOUR ORDER <<<<<<<<<

[('Cake', 5)]

 ......Updating Stock......


*** Stock after purchase ***
-----------------------------------

{'Bread': 20, 'Muffins': 50, 'Cake': 25, 'Cookies': 100}

 *** Stock value after last purchase ***
-----------------------------------------

{'Bread': 40, 'Muffins': 150, 'Cake': 62.5, 'Cookies': 50.0}
```

Graph:

Purchase details of 3 customers can be represented graphically as below:



Item	customer 1	customer 2	customer 3
Bread	5	1	3
Muffin	2	0	5
Cake	1	1	2
Cookies	0	5	10