UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Palmes, Lewis Clark L.

Instructor:
Engr. Maria Rizette H. Sayo

August, 09, 2025

I. Objectives

Introduction

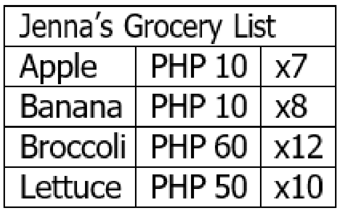
Array, in general, refers to an orderly arrangement of data elements. Array is a type of data structure that stores data elements in adjacent locations. Array is considered as linear data structure that stores elements of same data types. Hence, it is also called as a linear homogenous data structure.

This laboratory activity aims to implement the principles and techniques in:

- Writing algorithms using Array data structure
- Solve programming problems using dynamic memory allocation, arrays and pointers

II. Methods

Jenna's Grocery



Jenna's Grocery List		
Apple	PHP 10	x7
Banana	PHP 10	x8
Broccoli	PHP 60	x12
Lettuce	PHP 50	x10

Jenna wants to buy the following fruits and vegetables for her daily consumption. However, she needs to distinguish between fruit and vegetable, as well as calculate the sum of prices that she has to pay in total.

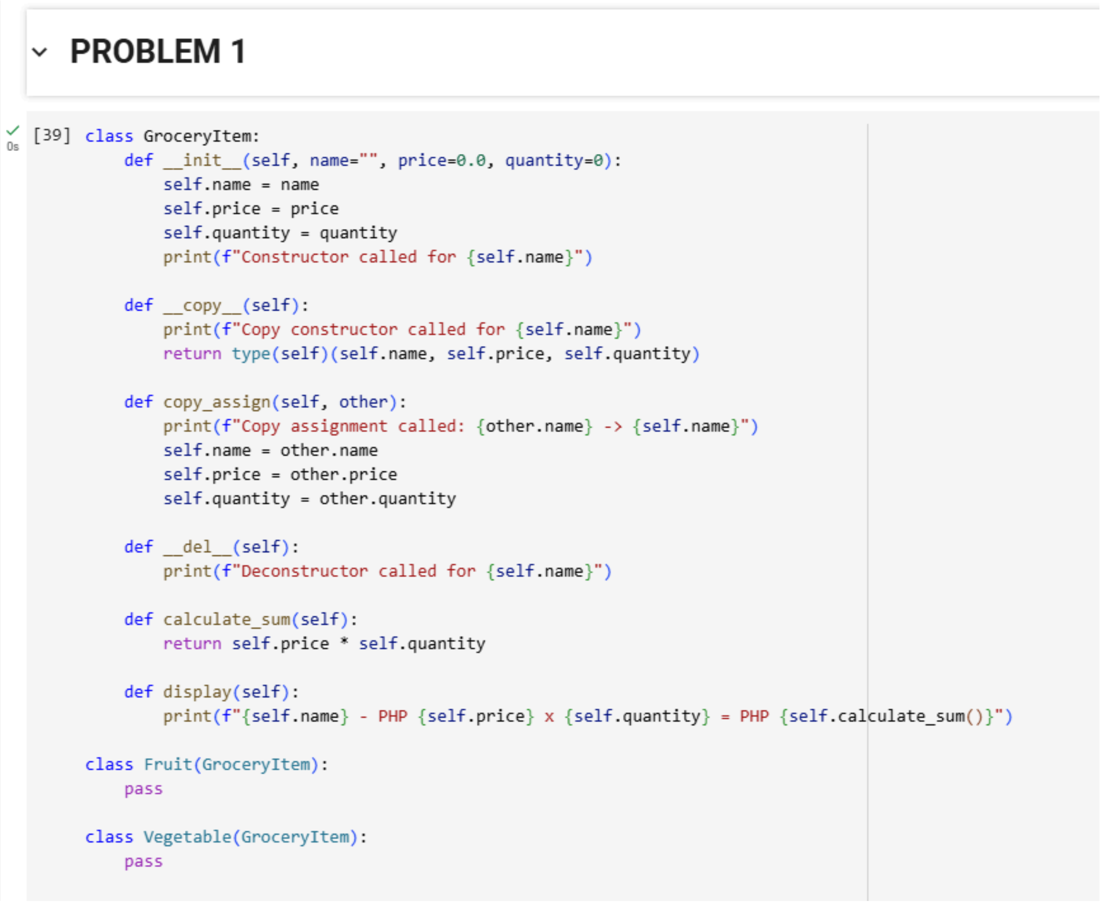
Problem 1: Create a class for the fruit and the vegetable classes. Each class must have a constructor, deconstructor, copy constructor and copy assignment operator. They must also have all relevant attributes (such as name, price and quantity) and functions (such as calculate sum) as presented in the problem description above.

Problem 2: Create an array GroceryList in the driver code that will contain all items in Jenna's Grocery List. You must then access each saved instance and display all details about the items.

Problem 3: Create a function TotalSum that will calculate the sum of all objects listed in Jenna's Grocery List.

Problem 4: Delete the Lettuce from Jenna's GroceryList list and de-allocate the memory assigned.

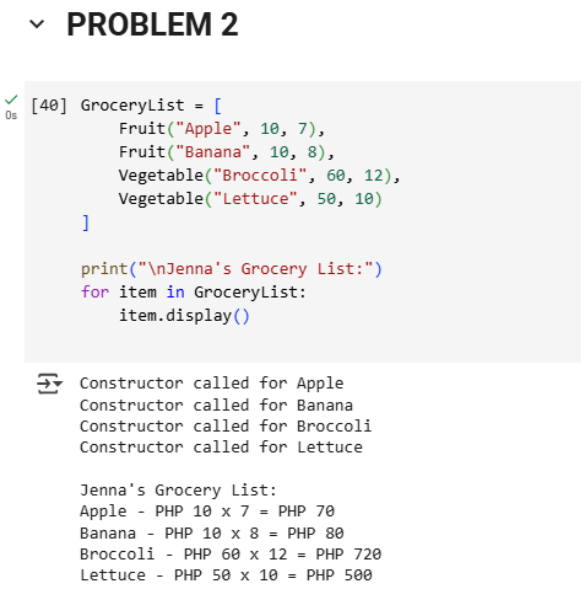
III. Results



```
class GroceryItem:
    def __init__(self, name="", price=0.0, quantity=0):
        self.name = name
        self.price = price
        self.quantity = quantity
        print(f"Constructor called for {self.name}")

    def __copy__(self):
        print(f"Copy constructor called for {self.name}")
        return type(self)(self.name, self.price, self.quantity)

    def copy_assign(self, other):
        print(f"Copy assignment called: {other.name} -> {self.name}")
        self.name = other.name
        self.price = other.price
        self.quantity = other.quantity

    def __del__(self):
        print(f"Deconstructor called for {self.name}")

    def calculate_sum(self):
        return self.price * self.quantity

    def display(self):
        print(f"{self.name} - PHP {self.price} x {self.quantity} = PHP {self.calculate_sum()}")

class Fruit(GroceryItem):
    pass

class Vegetable(GroceryItem):
    pass
```

Figure 1 Screenshot of program

This program involves designing object-oriented classes for Fruit and Vegetable with attributes like name, price, and quantity, plus methods for operations such as calculating the total cost. It also requires implementing constructor, copy constructor, copy assignment, and deconstructor equivalents in Python to mimic full class behavior.

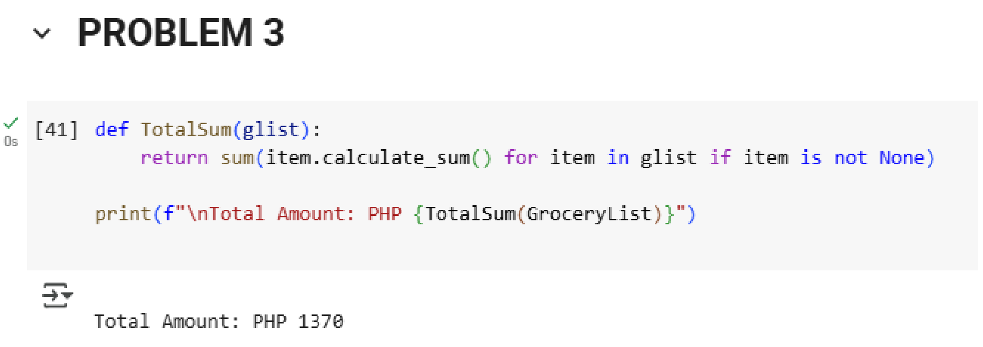


```
GroceryList = [
    Fruit("Apple", 10, 7),
    Fruit("Banana", 10, 8),
    Vegetable("Broccoli", 60, 12),
    Vegetable("Lettuce", 50, 10)
]

print("\nJenna's Grocery List:")
for item in GroceryList:
    item.display()
```

```
Constructor called for Apple
Constructor called for Banana
Constructor called for Broccoli
Constructor called for Lettuce

Jenna's Grocery List:
Apple - PHP 10 x 7 = PHP 70
Banana - PHP 10 x 8 = PHP 80
Broccoli - PHP 60 x 12 = PHP 720
Lettuce - PHP 50 x 10 = PHP 500
```

Figure 2 Screenshot of program

This program focuses on creating a GroceryList array (Python list) that stores different fruit and vegetable objects. The task is to loop through the list and display the details of each grocery item.



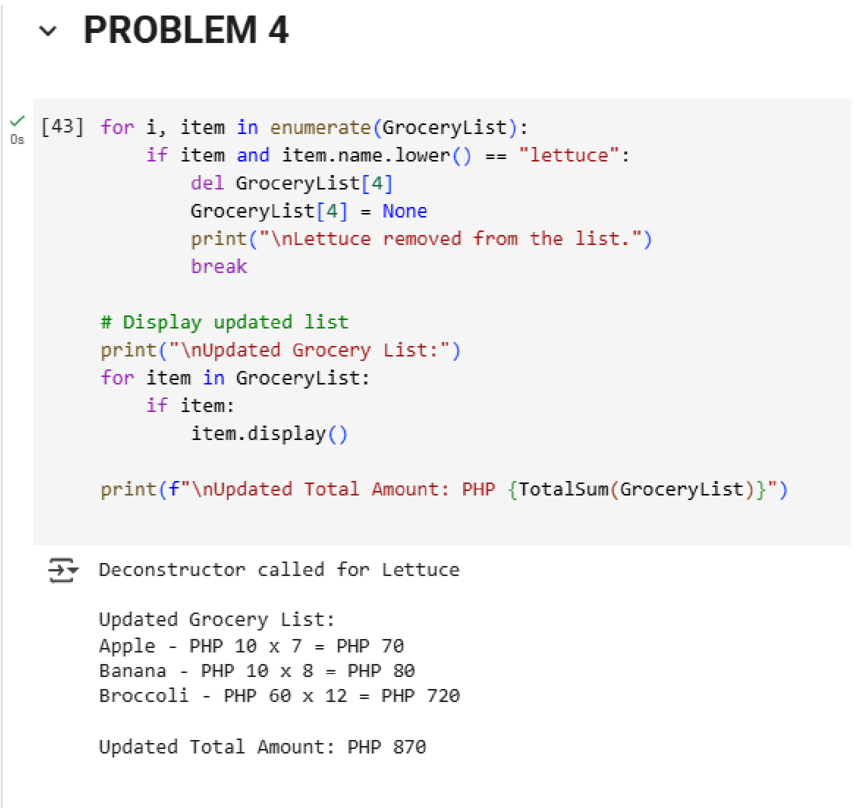
PROBLEM 3

```
[41] def TotalSum(glist):
         return sum(item.calculate_sum() for item in glist if item is not None)

     print(f"\nTotal Amount: PHP {TotalSum(GroceryList)}")
```

```
Total Amount: PHP 1370
```

Figure 3 Screenshot of program

This program requires writing a TotalSum function to calculate the combined cost of all items in the grocery list. It ensures you can process object attributes from a list and compute a total value dynamically.



PROBLEM 4

```
[43] for i, item in enumerate(GroceryList):
         if item and item.name.lower() == "lettuce":
             del GroceryList[4]
             GroceryList[4] = None
             print("\nLettuce removed from the list.")
             break

     # Display updated list
     print("\nUpdated Grocery List:")
     for item in GroceryList:
         if item:
             item.display()

     print(f"\nUpdated Total Amount: PHP {TotalSum(GroceryList)}")
```

```
Deconstructor called for Lettuce

Updated Grocery List:
Apple - PHP 10 x 7 = PHP 70
Banana - PHP 10 x 8 = PHP 80
Broccoli - PHP 60 x 12 = PHP 720

Updated Total Amount: PHP 870
```

Figure 4 Screenshot of program

This program is about removing the “Lettuce” item from the grocery list and ensuring the memory is freed. It demonstrates object deletion and updating the list to reflect the removed item.

IV. Conclusion

In conclusion, this activity showed how arrays and object-oriented programming can be used to solve a simple but practical problem like managing Jenna’s grocery list. By creating Fruit and Vegetable classes with the right functions and attributes, it became easy to organize the items, calculate the total cost, and even remove an item when needed. Using an array made storing and displaying the groceries straightforward, while the TotalSum function made it quick to see how much Jenna had to pay. Overall, the task was a hands-on way to practice structuring data, working with objects, and managing information in a program.

References

[1] Co Arthur O.. "University of Caloocan City Computer Engineering Department Honor Code," UCC-CpE Departmental Policies, 2020.

[2]*Google Colab*. (n.d.). https://colab.research.google.com/drive/1gVKoNs8sw8Z48CkD_k84Wt-WjJPve3x0#scrollTo=fxt186lE08au

[3]Lewis-Clark-Palmes. (n.d.). *CPE-201L-DSA-2-A/LAB_4.ipynb at main · Lewis-Clark-Palmes/CPE-201L-DSA-2-A*. GitHub. https://github.com/Lewis-Clark-Palmes/CPE-201L-DSA-2-A/blob/main/LAB_4.ipynb