UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Sumel, Hendrix Nathan L.

Instructor:
Engr. Maria Rizette H. Sayo

August 9, 2025

I. Objectives

Introduction

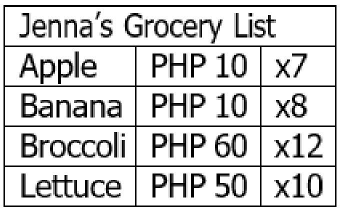
Array, in general, refers to an orderly arrangement of data elements. Array is a type of data structure that stores data elements in adjacent locations. Array is considered as linear data structure that stores elements of same data types. Hence, it is also called as a linear homogenous data structure.

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

Problem 1 – Create Fruit and Vegetable Classes

- Start
- Define a Fruit class with attributes: name, price, quantity
- Define a Vegetable class with attributes: name, price, quantity
- Inside each class, create a method calculate_sum() = price × quantity
- End

Problem 2 – Store Grocery Items in a List

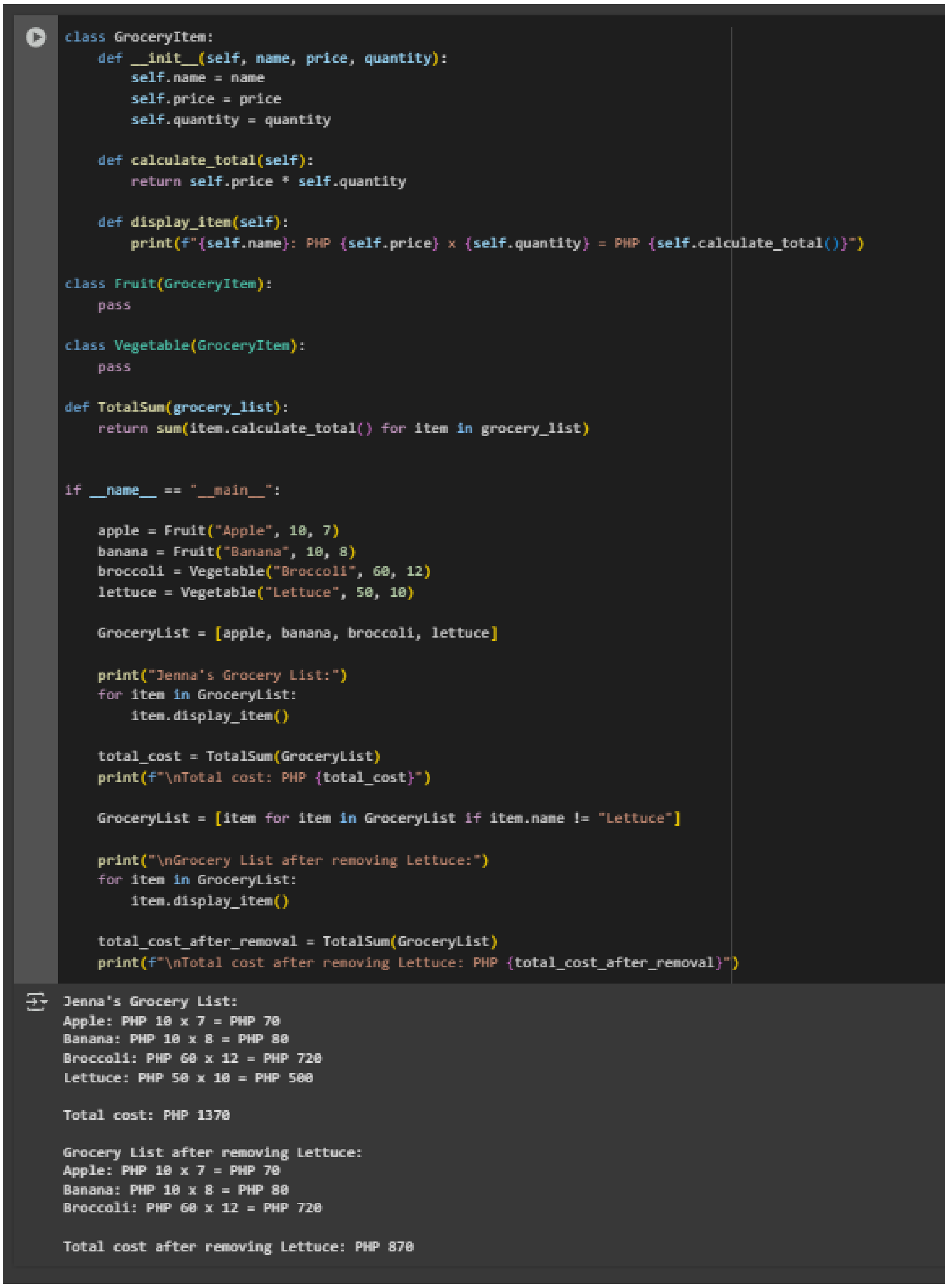
- Start
- Initialize an empty list grocery_list[]
- Create objects for each fruit and vegetable
- Append all objects to the grocery_list
- For each object in the list, display name, price, quantity, and total cost
- End

Problem 3 – Compute Total Sum of All Items

- Start
- Define a function total_sum(grocery_list)
- Initialize variable total = 0
- For each object in grocery_list, add object.calculate_sum() to total
- Return or print total
- End

Problem 4 – Remove Lettuce from the List

- Start
- Traverse grocery_list
- Find the item where name == "Lettuce"
- Remove that item from the list using remove()
- End



```python
class GroceryItem:
    def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

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

    def display_item(self):
        print(f"{self.name}: PHP {self.price} x {self.quantity} = PHP {self.calculate_total()}")

class Fruit(GroceryItem):
    pass

class Vegetable(GroceryItem):
    pass

def TotalSum(grocery_list):
    return sum(item.calculate_total() for item in grocery_list)


if __name__ == "__main__":

    apple = Fruit("Apple", 10, 7)
    banana = Fruit("Banana", 10, 8)
    broccoli = Vegetable("Broccoli", 60, 12)
    lettuce = Vegetable("Lettuce", 50, 10)

    GroceryList = [apple, banana, broccoli, lettuce]

    print("Jenna's Grocery List:")
    for item in GroceryList:
        item.display_item()

    total_cost = TotalSum(GroceryList)
    print(f"\nTotal cost: PHP {total_cost}")

    GroceryList = [item for item in GroceryList if item.name != "Lettuce"]

    print("\nGrocery List after removing Lettuce:")
    for item in GroceryList:
        item.display_item()

    total_cost_after_removal = TotalSum(GroceryList)
    print(f"\nTotal cost after removing Lettuce: PHP {total_cost_after_removal}")
```

```
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

Total cost: PHP 1370

Grocery List after removing Lettuce:
Apple: PHP 10 x 7 = PHP 70
Banana: PHP 10 x 8 = PHP 80
Broccoli: PHP 60 x 12 = PHP 720

Total cost after removing Lettuce: PHP 870
```

The program uses Object-Oriented Programming in Python to represent Jenna’s grocery items. A parent class called GroceryItem was created with attributes for name, price, and quantity, along with methods to calculate the total cost and display the details of each item. Two subclasses, Fruit and Vegetable, inherit from this class, showing that both fruits and vegetables are types of grocery items. The program then displays each item with its total cost, computes the overall sum using the TotalSum function, and prints the result. Afterwards, Lettuce is removed from the list, and the updated grocery list and new total cost are displayed. This program shows how Python classes and lists can be used together to solve real-life problems in a structured and simple way.

IV. Conclusion

This lab activity taught me how to use Python classes and lists to represent a simple grocery list. I created a main class and used inheritance for fruits and vegetables so I could store their name, price, and quantity. From there, I was able to show the details of each item, add up the total cost, and even remove one item from the list. At first it was a bit confusing to connect everything, but once I got the flow, the code made more sense. Overall, I learned how arrays (lists), functions, and OOP can work together to solve problems in a more organized way.

References

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

W3Schools. *Python Classes and Objects.* https://www.w3schools.com/python/python_classes.asp

Programiz. *Python List (Array).* https://www.programiz.com/python-programming/list