UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Poliño, Justine

Instructor:
Engr. Maria Rizette H. Sayo

August 17, 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

This program is acting as a smart assistant for Jenna. First it starts by creating a digital list of items, storing each of fruit and vegetable's name, quantity, and price. The items display are neatly in a receipt like table with their individual costs and their categories. Her total is automatically being calculated from her spending and elimating the manual calculation. When Jenna wants to change something (ex. removing a vegetables or fruits), the list is being update instantly, that will recalculate her new total. Grocery management is made easy by the straightforward, well-organized structure, which combines adjustable alterations with clear information, simulating real shopping without the trouble.

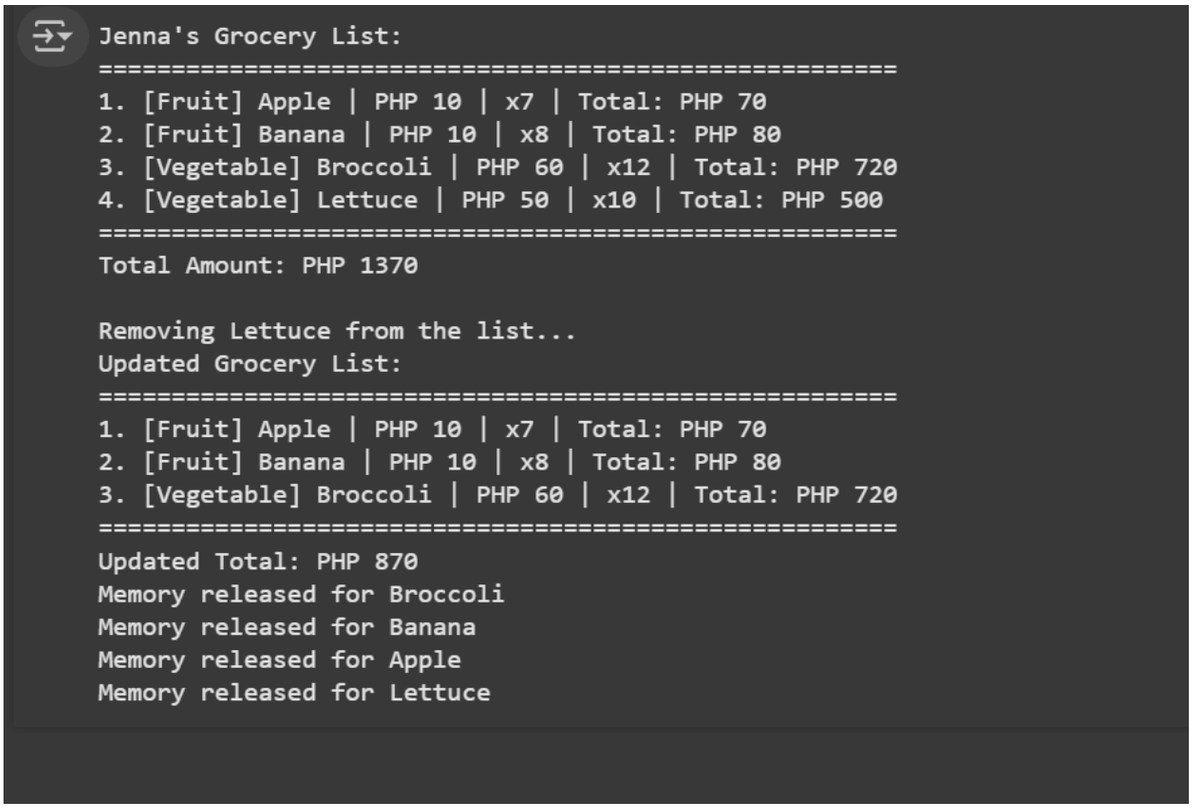

Figure 1 Screenshot of the output

SOURCECODE

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

    def __del__(self):
        print(f"Memory released for {self.name}")

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

    def __str__(self):
        return f"{self.name} | PHP {self.price} | x{self.quantity} | Total: PHP {self.total()}"

class Fruit(GroceryItem):
    def __str__(self):
        return f"[Fruit] {super().__str__()}"

class Vegetable(GroceryItem):
    def __str__(self):
        return f"[Vegetable] {super().__str__()}"

def main():
    # Problem 1 & 2: Create classes and array GroceryList
    items = [Fruit("Apple", 10, 7), Fruit("Banana", 10, 8),
             Vegetable("Broccoli", 60, 12), Vegetable("Lettuce", 50, 10)]

    # Problem 2: Display all items in the array
    print("Jenna's Grocery List:\n" + "=" * 55)
    for i, item in enumerate(items, 1):
        print(f"{i}. {item}")

    # Problem 3: Calculate total sum of all items
    total = sum(item.total() for item in items)
    print("=" * 55 + f"\nTotal Amount: PHP {total}\n")

    # Problem 4: Delete Lettuce from the list
    print("Removing Lettuce from the list...")
    items = [item for item in items if item.name != "Lettuce"]

    # Display updated list after deletion
    print("Updated Grocery List:\n" + "=" * 55)
    for i, item in enumerate(items, 1):
        print(f"{i}. {item}")

    # Calculate new total after deletion
    new_total = sum(item.total() for item in items)
    print("=" * 55 + f"\nUpdated Total: PHP {new_total}")

if __name__ == "__main__":
    main()
```

Figure 1 Screenshot of the source code

ALGORITHM

1. Define GroceryItem, Fruit, and Vegetable classes with:
   - Constructor (name, price, quantity)
   - total() method (price × quantity)
   - __str__() for display formatting
2. Create array with 4 grocery items:
   [Fruit("Apple",10,7), Fruit("Banana",10,8),
   Vegetable("Broccoli",60,12), Vegetable("Lettuce",50,10)]
3. Display original list with item details
4. Calculate and display total sum (Problem 3)
5. Remove "Lettuce" from array (Problem 4)
6. Display updated list without Lettuce
7. Calculate and display new total sum

IV. Conclusion

This grocery list program operates similarly to a helpful digital assistant; it organizes the data, performs calculations for her, and modifies the code if she changes what she thinks. The program simplifies and makes grocery management easy to understand by explicitly outlining each process, from making the list to updating it It serves as a real-world illustration of how simple coding may resolve common issues and transform what could be into something quick and simple. As a beginner, I simply value simplicity, and this project demonstrates how even a small amount of structure can greatly simplify life.

References

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