UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Bron, Jhustine A.

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

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

    def __del__(self):
        pass

    def copy(self):
        return Fruit(self.name, self.price, self.quantity)

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

class Vegetable:
    def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    def __del__(self):
        pass

    def copy(self):
        return Vegetable(self.name, self.price, self.quantity)

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

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

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

print("Jenna's Grocery List\n")
for item in GroceryList:
    print(f"{item.name}: PHP {item.price} x {item.quantity} = PHP {item.calculate_sum()}")

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

print("\nRemoving Lettuce:")
for item in GroceryList:
    if item.name == "Lettuce":
        print(f"Destructor called for {item.name}")
        GroceryList.remove(item)
        break
print(f"After removing the Lettuce using destructor: PHP {TotalSum(GroceryList)}")
```

Output:

```
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: PHP 1370

Removing Lettuce:
Destructor called for Lettuce
After removing the Lettuce using destructor: PHP 870
```

Figure 1. Screenshot of the program

Colab link:
https://colab.research.google.com/drive/1lQNoRlC6QS14_SW88ROpdgdnWHn6AGSP#scrollTo=DSLw8T432UPb

Insights:

In this task, I created two classes, Fruit and Vegetable, each with a constructor, destructor, copy constructor, and a method to calculate the total cost. I then made a grocery list using objects from both classes, displayed each item's details, and calculated the overall total using a TotalSum function. Finally, I removed the lettuce item from the list and simulated calling its destructor. This code helped me understand how to work with classes, object creation and deletion, and basic list operations in Python, while also practicing how to organize and reuse code effectively.

IV. Conclusion

This activity helped me practice using arrays and classes to organize data and solve problems. I created Fruit and Vegetable classes with constructors, destructors, and other methods, stored them in a GroceryList, calculated the total cost, and removed an item. It improved my understanding of arrays, object-oriented programming, and basic memory management.