UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Directo, Hannah Thea B.

Instructor:
Engr. Maria Rizette H. Sayo

August 13, 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 total_price(self):
        return self.price * self.quantity

    def display(self):
        print(f"[Fruit] {self.name} - PHP {self.price} x {self.quantity} = PHP {self.total_price()}")

    def __del__(self):
        pass

    def copy(self):
        pass

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

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

    def display(self):
        print(f"[Vegetable] {self.name} - PHP {self.price} x {self.quantity} = PHP {self.total_price()}")

    def __del__(self):
        pass

    def copy(self):
        pass
```

Figure 1 Screenshot of Source code

```python
def sum_total_price(items):
    return sum(item.total_price() for item in items)

if __name__ == "__main__":

    # Problem 1 & 2: Create the grocery list
    fruits = [
        Fruit("Apple", 10, 7),
        Fruit("Banana", 10, 8)
    ]

    vegetables = [
        Vegetable("Broccoli", 60, 12),
        Vegetable("Lettuce", 50, 10)
    ]

    GroceryList = fruits + vegetables

    # Display the full grocery list
    print("____Jenna's Grocery List____\n")
    for item in GroceryList:
        item.display()

    # Problem 3: Show total cost
    print(f"\nTotal Price: PHP {sum_total_price(GroceryList)}")
    print("-----------------------------------------")

    # Problem 4: Delete Lettuce using simple method and destructor
    print("Removing Lettuce from the Grocery List...\n")
    for item in GroceryList:
        if type(item).__name__ == "Vegetable" and item.name.lower() == "lettuce":
            print("Calling destructor for Lettuce...\n")
            GroceryList.remove(item)
            del item
            break

    # Updated grocery list
    print("_____Updated Grocery List_____\n")
    for item in GroceryList:
        item.display()

    print(f"\nTotal Price: PHP {sum_total_price(GroceryList)}")
```

Figure 2 Screenshot of Source code

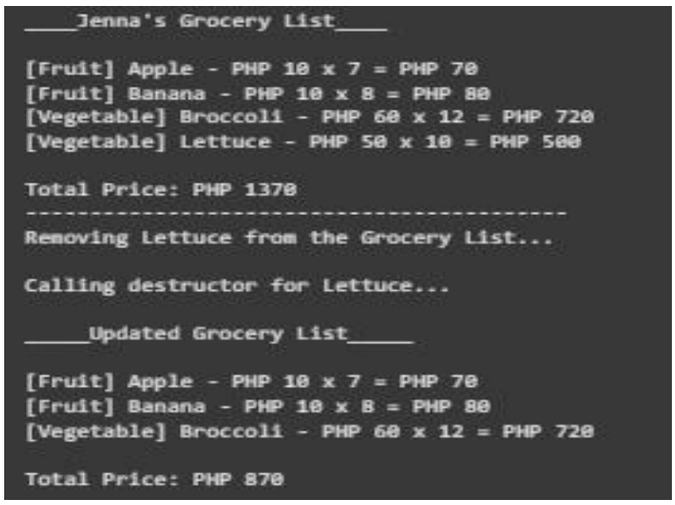

Figure 2 Screenshot of Output

Refer to this link:
https://colab.research.google.com/github/hannahdirecto/CPE-201L-DSA-2-A/blob/main/LAb_Act_4.ipynb#scrollTo=B-ZLaEZrxslB

Insights: The program has two classes, Fruit and Vegetable, which share the same attributes: name, price, and quantity. All these items are stored together in one list called GroceryList, making it easy to manage them as a group. The program includes a function that calculates the total cost by multiplying the price and quantity of each item and then adding them all up. It also neatly displays each item's details. Additionally, the program shows how to remove an item, specifically "Lettuce", from the list. After deleting the lettuce, it updates the grocery list and calculates the new total price. The code uses del to delete the item from memory, simulating memory cleanup.

IV. Conclusion

In this activity, we learned how to use arrays and classes to organize and manage data. We created fruit and vegetable objects, stored them in one list, and used functions to show their details and calculate the total cost. We also practiced how to remove an item from the list and clean up memory using del. This activity helped us understand how arrays and basic programming techniques work together to solve real-life problems.