UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Barbas, Steven Jade P.

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

class Fruit(GroceryItem):
    def __init__(self, name, price, quantity):
        super().__init__(name, price, quantity)

    def __del__(self):
        pass

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

    def display(self):
        subtotal = self.calculate_sum()
        return f"{'Fruit':<12} {self.name:<12} {self.price:<10} {self.quantity:<10} {subtotal:<10}"

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

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

    def __del__(self):
        pass

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

    def display(self):
        subtotal = self.calculate_sum()
        return f"{'Vegetable':<12} {self.name:<12} {self.price:<10} {self.quantity:<10} {subtotal:<10}"

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

def TotalSum(grocery_list):
    total = 0
    for item in grocery_list:
        total += item.calculate_sum()
    return total
```

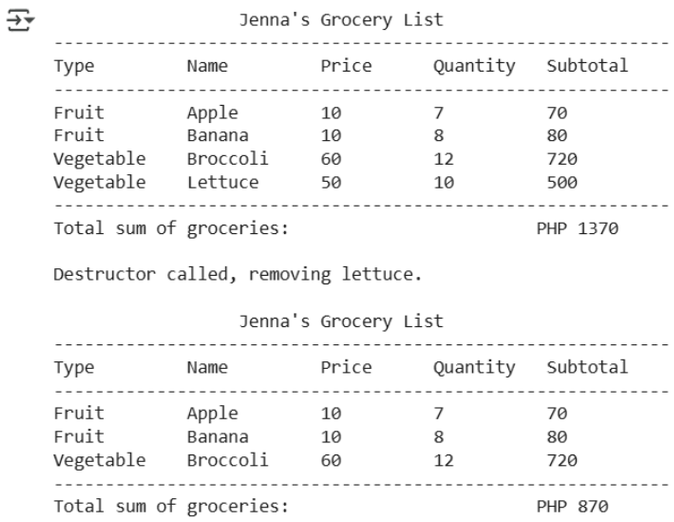
Figure 1 Screenshot of program

```python
# --- Main program ---
apple = Fruit("Apple", 10, 7)
banana = Fruit("Banana", 10, 8)
broccoli = Vegetable("Broccoli", 60, 12)
lettuce = Vegetable("Lettuce", 50, 10)
grocery_list = [apple, banana, broccoli, lettuce]

print("                  Jenna's Grocery List")
print("-------------------------------------------------------------")
print(f"{'Type':<12} {'Name':<12} {'Price':<10} {'Quantity':<10} {'Subtotal':<10}")
print("-------------------------------------------------------------")
for item in grocery_list:
    print(item.display())
print("-------------------------------------------------------------")
total = TotalSum(grocery_list)
print(f"{'Total sum of groceries:':<46} PHP {total}\n")


# Remove lettuce
print("Destructor called, removing lettuce.\n")
grocery_list.remove(lettuce)
del lettuce

print("                  Jenna's Grocery List")
print("-------------------------------------------------------------")
print(f"{'Type':<12} {'Name':<12} {'Price':<10} {'Quantity':<10} {'Subtotal':<10}")
print("-------------------------------------------------------------")
for item in grocery_list:
    print(item.display())
print("-------------------------------------------------------------")
total = TotalSum(grocery_list)
print(f"{'Total sum of groceries:':<46} PHP {total}\n")
```

Figure 2 Screenshot of program

This Python program uses classes and lists to manage a grocery inventory. The base class, Grocery Item, holds attributes such as name, price, and quantity, while the Fruit and Vegetable classes inherit these properties. The program creates a list with four items (Apple, Banana, Broccoli, and Lettuce), displays their details, and calculates the total cost. Afterward, Lettuce is removed from the list, the updated items are shown, and the total is recalculated. It also demonstrates copying an item. Overall, the program illustrates how to store, update, and delete objects in a list while applying fundamental object-oriented programming concepts.

Output :



```
                Jenna's Grocery List
------------------------------------------------------------
Type         Name         Price      Quantity   Subtotal
------------------------------------------------------------
Fruit        Apple        10         7          70
Fruit        Banana       10         8          80
Vegetable    Broccoli     60         12         720
Vegetable    Lettuce      50         10         500
------------------------------------------------------------
Total sum of groceries:                        PHP 1370

Destructor called, removing lettuce.

                Jenna's Grocery List
------------------------------------------------------------
Type         Name         Price      Quantity   Subtotal
------------------------------------------------------------
Fruit        Apple        10         7          70
Fruit        Banana       10         8          80
Vegetable    Broccoli     60         12         720
------------------------------------------------------------
Total sum of groceries:                        PHP 870
```

Figure 3 Screenshot of program output

IV. Conclusion

This activity showed how arrays can be used in Python to handle many objects at once. By adding, showing, removing, and copying grocery items, we learned how arrays work together with object-oriented programming. It also showed that arrays make it easier to keep related data organized and updated in one place. By making a grocery list, we practiced looping through arrays, finding totals, and changing items. Overall, this activity helped us understand arrays better and introduced ideas like inheritance, constructors, and object deletion in Python.

References

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