



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 4

---

# Arrays

---

*Submitted by:*  
Dispo, Lei Andrew T.

*Instructor:*  
Engr. Maria Rizette H. Sayo

10, 11, 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

## ALGORITHM

1. Start the program.
2. Create a class “GroceryItem” with attributes: name, price, and quantity. Include functions to calculate the total sum and display the item details.
3. Create two subclasses “Fruit” and “Vegetable” that inherit from the GroceryItem class.
4. Create a grocery list containing several fruits and vegetables with their price and quantity.
5. Display all the items in the grocery list.
6. Compute and display the total sum of all grocery items.
7. Remove Lettuce from the grocery list.
8. Display the updated grocery list.
9. Compute and display the updated total sum after deletion.
10. End the program.

## CODE

### 1) Creating a Class

```
# Problem 1: Create Fruit and Vegetable Classes
class GroceryItem: 2 usages
    def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    def calculate_sum(self): 2 usages (1 dynamic)
        return self.price * self.quantity

    def display(self): 2 usages (1 dynamic)
        print(f"{self.name} - PHP {self.price} x {self.quantity} = PHP {self.calculate_sum()}")

class Fruit(GroceryItem): 2 usages
    def __init__(self, name, price, quantity):
        super().__init__(name, price, quantity)

class Vegetable(GroceryItem): 2 usages
    def __init__(self, name, price, quantity):
        super().__init__(name, price, quantity)
```

## 2) Creating an Array

```
# Problem 2: Create GroceryList and Display All Items
def create_grocery_list(): 1 usage
    GroceryList = [
        Fruit( name: "Apple", price: 10, quantity: 7),
        Fruit( name: "Banana", price: 10, quantity: 8),
        Vegetable( name: "Broccoli", price: 60, quantity: 12),
        Vegetable( name: "Lettuce", price: 50, quantity: 10)
    ]
    return GroceryList
```

```
def main(): 1 usage
    # Problem 2 completion
    GroceryList = create_grocery_list()
```

## 3) Create a Total Sum

```
# Problem 3: Create Function TotalSum
def TotalSum(grocery_list): 2 usages
    total = 0
    for item in grocery_list:
        total += item.calculate_sum()
    return total
```

## 4) Deleting an element

```
# Problem 4: Delete Lettuce and Display Updated List

def delete_lettuce(grocery_list): 1 usage
    updated_list = [item for item in grocery_list if item.name != "Lettuce"]
    return updated_list
```

```
# Problem 4 completion
GroceryList = delete_lettuce(GroceryList)

print("After deleting Lettuce:")
for item in GroceryList:
    item.display()
print(f"Total Sum = PHP {TotalSum(GroceryList)}")
```

## PROGRAM OUTPUT

```
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 sum: PHP 1370

After deleting Lettuce:
Apple - PHP 10 x 7 = PHP 70
Banana - PHP 10 x 8 = PHP 80
Broccoli - PHP 60 x 12 = PHP 720
Total Sum = PHP 870
```

## IV. Conclusion

The program successfully organizes Jenna's grocery list using classes and objects. It applies inheritance by creating Fruit and Vegetable subclasses from the GroceryItem class, each containing details like name, price, and quantity. All grocery items are stored as elements in an array for easy management. The program calculates the total cost, removes an item (Lettuce), and updates the list and total.

## References

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