UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Maringal, Czer Justine D.

Instructor:
Engr. Maria Rizette H. Sayo

August 9, 2025

I. Objectives

Introduction

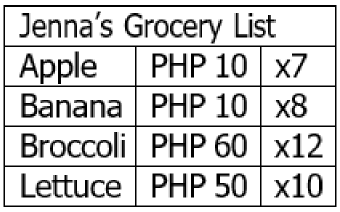
Array, in general, refers to an orderly arrangement of data elements. Array is a type of data structure that stores data elements in adjacent locations. Array is considered as linear data structure that stores elements of same data types. Hence, it is also called as a linear homogenous data structure.

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

<u>Jenna's Grocery</u>



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.

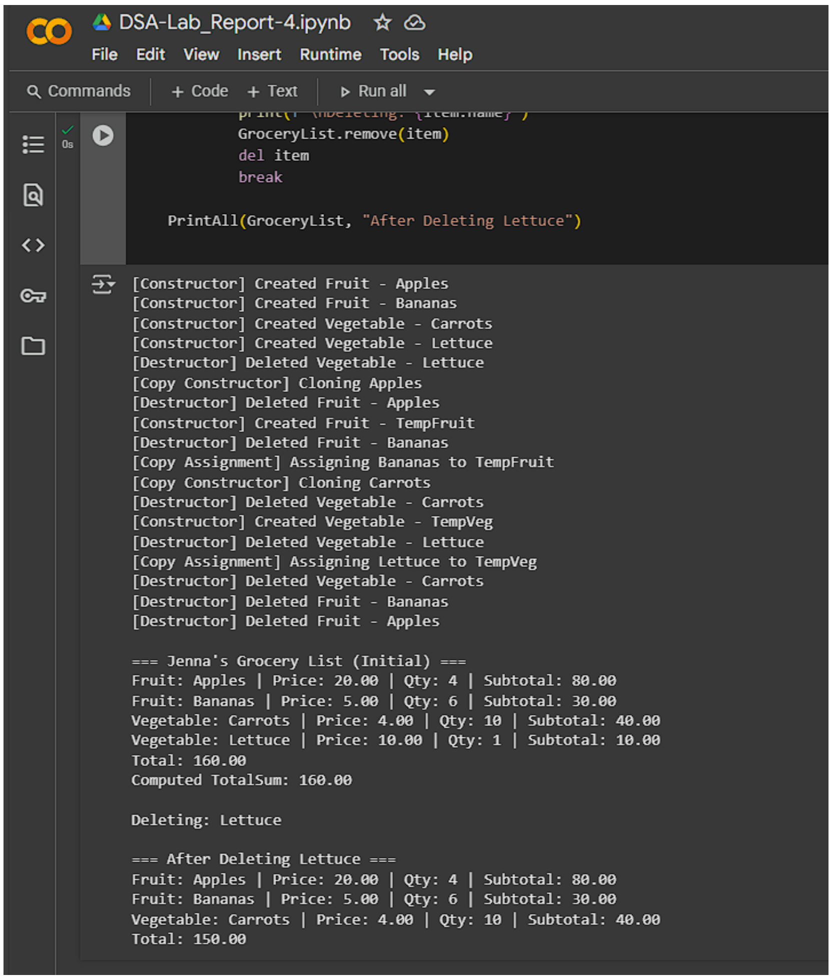
Problem 1: Create a class for the fruit and the vegetable classes. Each class must have a constructor, deconstructor, copy constructor and copy assignment operator. They must also have all relevant attributes (such as name, price and quantity) and functions (such as calculate sum) as presented in the 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



```
            print(f"\nDeleting: {item.name}")
            GroceryList.remove(item)
            del item
            break

    PrintAll(GroceryList, "After Deleting Lettuce")
```

```
[Constructor] Created Fruit - Apples
[Constructor] Created Fruit - Bananas
[Constructor] Created Vegetable - Carrots
[Constructor] Created Vegetable - Lettuce
[Destructor] Deleted Vegetable - Lettuce
[Copy Constructor] Cloning Apples
[Destructor] Deleted Fruit - Apples
[Constructor] Created Fruit - TempFruit
[Destructor] Deleted Fruit - Bananas
[Copy Assignment] Assigning Bananas to TempFruit
[Copy Constructor] Cloning Carrots
[Destructor] Deleted Vegetable - Carrots
[Constructor] Created Vegetable - TempVeg
[Destructor] Deleted Vegetable - Lettuce
[Copy Assignment] Assigning Lettuce to TempVeg
[Destructor] Deleted Vegetable - Carrots
[Destructor] Deleted Fruit - Bananas
[Destructor] Deleted Fruit - Apples

=== Jenna's Grocery List (Initial) ===
Fruit: Apples | Price: 20.00 | Qty: 4 | Subtotal: 80.00
Fruit: Bananas | Price: 5.00 | Qty: 6 | Subtotal: 30.00
Vegetable: Carrots | Price: 4.00 | Qty: 10 | Subtotal: 40.00
Vegetable: Lettuce | Price: 10.00 | Qty: 1 | Subtotal: 10.00
Total: 160.00
Computed TotalSum: 160.00

Deleting: Lettuce

=== After Deleting Lettuce ===
Fruit: Apples | Price: 20.00 | Qty: 4 | Subtotal: 80.00
Fruit: Bananas | Price: 5.00 | Qty: 6 | Subtotal: 30.00
Vegetable: Carrots | Price: 4.00 | Qty: 10 | Subtotal: 40.00
Total: 150.00
```

This creates Fruit and Vegetable classes derived from a base GroceryItem class to manage grocery items with attributes like name, price, and quantity. It includes a constructor to create objects, a destructor to delete them, a copy constructor to clone objects, and a copy assignment method to copy values between objects. The program calculates the subtotal for each item, stores items in a grocery list, computes the total cost, and allows deletion of specific items, simulating real-world grocery management using object-oriented programming concepts.

1. GroceryItem class (base)

- Stores name, price, and quantity.
- __init__() → Constructor: runs when an object is created.
- __del__() → Destructor: runs when an object is deleted.
- copy_constructor() → Creates a deep copy of the object.
- assign() → Copies values from another object (like copy assignment in C++).
- calculate_sum() → Returns total cost (price × quantity).
- display() → Prints item details.

2. Fruit & Vegetable classes

- Inherit from GroceryItem without adding new behavior (specialized types).

3. Utility functions

- TotalSum() → Adds up total cost of all items in a list.
- PrintAll() → Prints all items and the grand total.

4. Main program flow

- Creates several fruit and vegetable objects.
- Demonstrates copy constructor and copy assignment.
- Stores them in a grocery list and displays details.
- Calculates total price of all items.
- Deletes "Lettuce" from the list and updates the display.

IV. Conclusion

In conclusion, this program demonstrates how object-oriented programming concepts can be applied in Python to manage grocery items effectively. Using classes, inheritance, and methods, it allows the creation, copying, assignment, and deletion of objects, while encapsulating data like name, price, and quantity. The program also shows how to calculate subtotals for individual items and the total cost for a list, simulating real-world grocery management. Overall, it highlights the practical use of constructors, destructors, copy operations, and lists in organizing and manipulating data efficiently.

References

[1] Python Software Foundation. “Classes in Python,” Python Documentation,
https://docs.python.org/3/tutorial/classes.html , 2025.

[2] Python Software Foundation. “Data Model — __del__ method,” Python Documentation,
https://docs.python.org/3/reference/datamodel.html#object.__del __, 2025.

[3] Python Software Foundation. “Copy Module,” Python Documentation,
https://docs.python.org/3/library/copy.html , 2025.

[4] Python Software Foundation. “Inheritance in Python,” Python Documentation,
https://docs.python.org/3/tutorial/classes.html#inheritance , 2025.

[5] GeeksforGeeks. “Python OOPS Concepts,
” https://www.geeksforgeeks.org/python/python-oops-concepts/, 2025.

[6] GeeksforGeeks. “Python | Maintaining Grocery list,
” https://www.geeksforgeeks.org/python/python-maintaining-grocery-list/ , 2025.