UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Talagtag Mark Angel T.

Instructor:
Engr. Maria Rizette H. Sayo

August 30, 2025

I. Objectives

Introduction

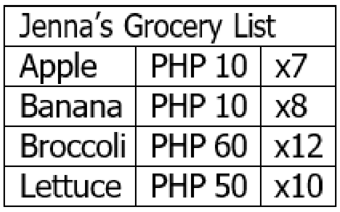
Array, in general, refers to an orderly arrangement of data elements. Array is a type of data structure that stores data elements in adjacent locations. Array is considered as linear data structure that stores elements of same data types. Hence, it is also called as a linear homogenous data structure.

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.

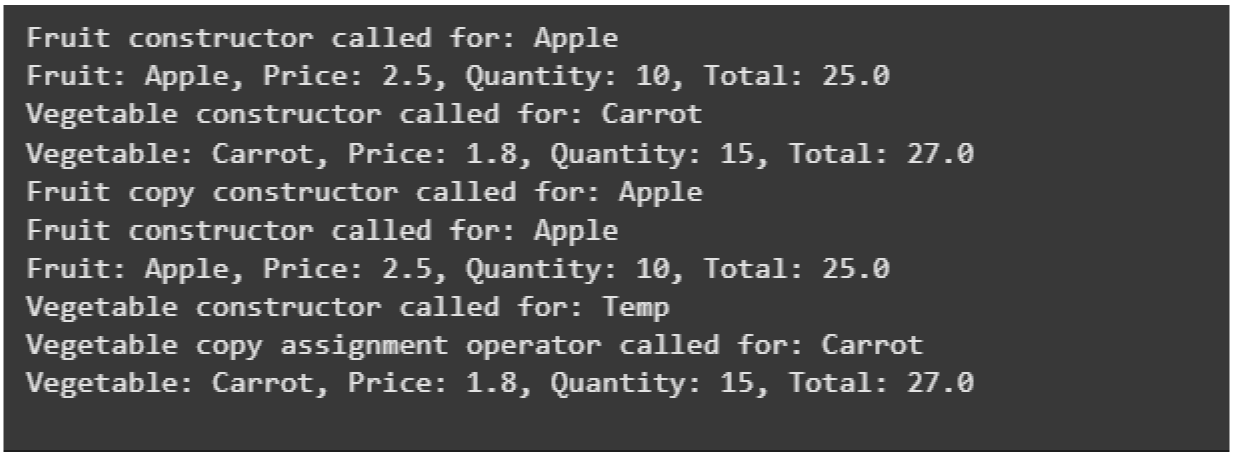
Problem 1: Create a class for the fruit and the vegetable classes. Each class must have a constructor, deconstructor, copy constructor and copy assignment operator. They must also have all relevant attributes (such as name, price and quantity) and functions (such as calculate sum) as presented in the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

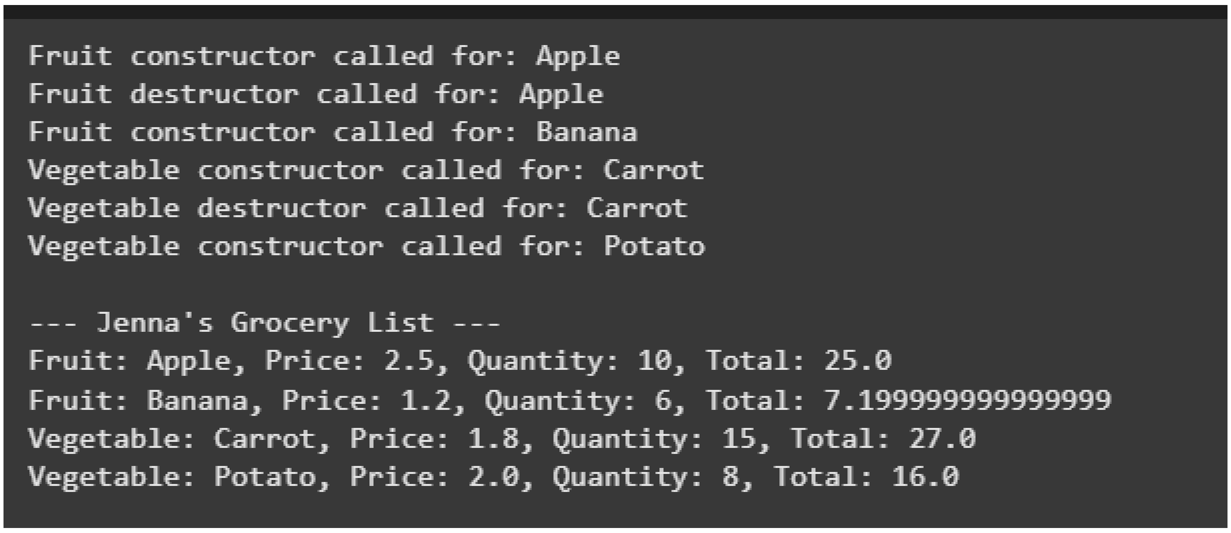


```
Fruit constructor called for: Apple
Fruit: Apple, Price: 2.5, Quantity: 10, Total: 25.0
Vegetable constructor called for: Carrot
Vegetable: Carrot, Price: 1.8, Quantity: 15, Total: 27.0
Fruit copy constructor called for: Apple
Fruit constructor called for: Apple
Fruit: Apple, Price: 2.5, Quantity: 10, Total: 25.0
Vegetable constructor called for: Temp
Vegetable copy assignment operator called for: Carrot
Vegetable: Carrot, Price: 1.8, Quantity: 15, Total: 27.0
```

Constructors are called when new objects are created (Apple, Carrot, Temp).

Copy constructor (copy) is called when making a copy (Apple Apple copy).

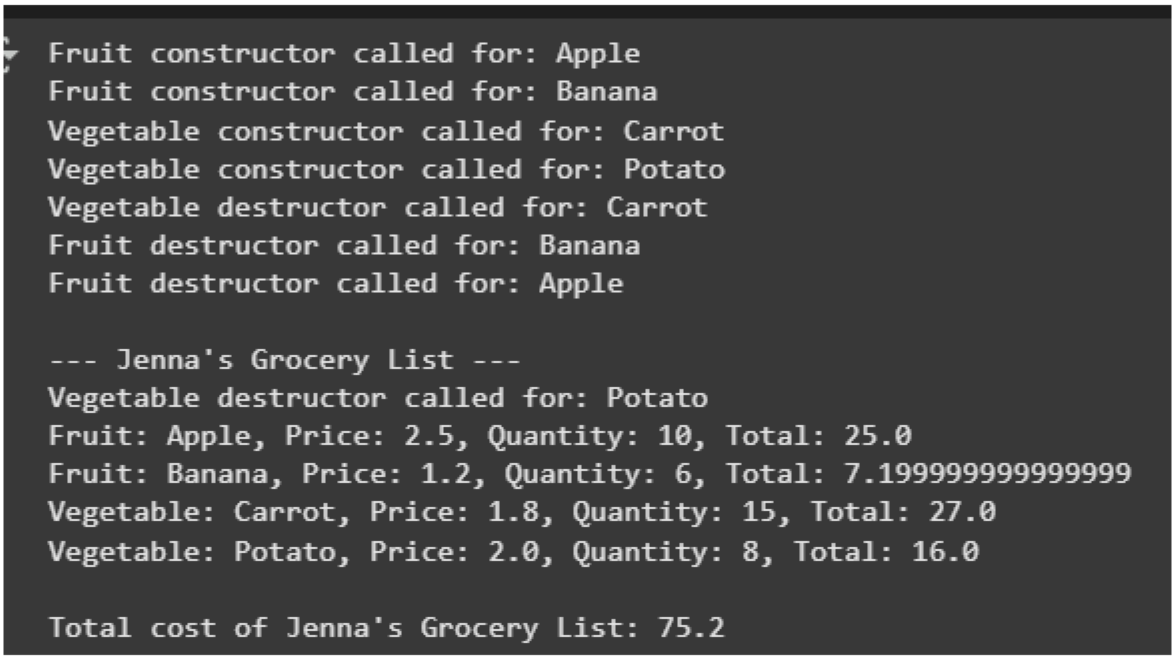
Copy assignment (copy_assign) is used to overwrite one object’s attributes with another’s (Temp Carrot).



```
Fruit constructor called for: Apple
Fruit destructor called for: Apple
Fruit constructor called for: Banana
Vegetable constructor called for: Carrot
Vegetable destructor called for: Carrot
Vegetable constructor called for: Potato

--- Jenna's Grocery List ---
Fruit: Apple, Price: 2.5, Quantity: 10, Total: 25.0
Fruit: Banana, Price: 1.2, Quantity: 6, Total: 7.199999999999999
Vegetable: Carrot, Price: 1.8, Quantity: 15, Total: 27.0
Vegetable: Potato, Price: 2.0, Quantity: 8, Total: 16.0
```

In Python, destructors (__del__) run when the reference count for an object drops to zero.

If you create an object but do not store it properly in a list or variable, it may get deleted instantly.

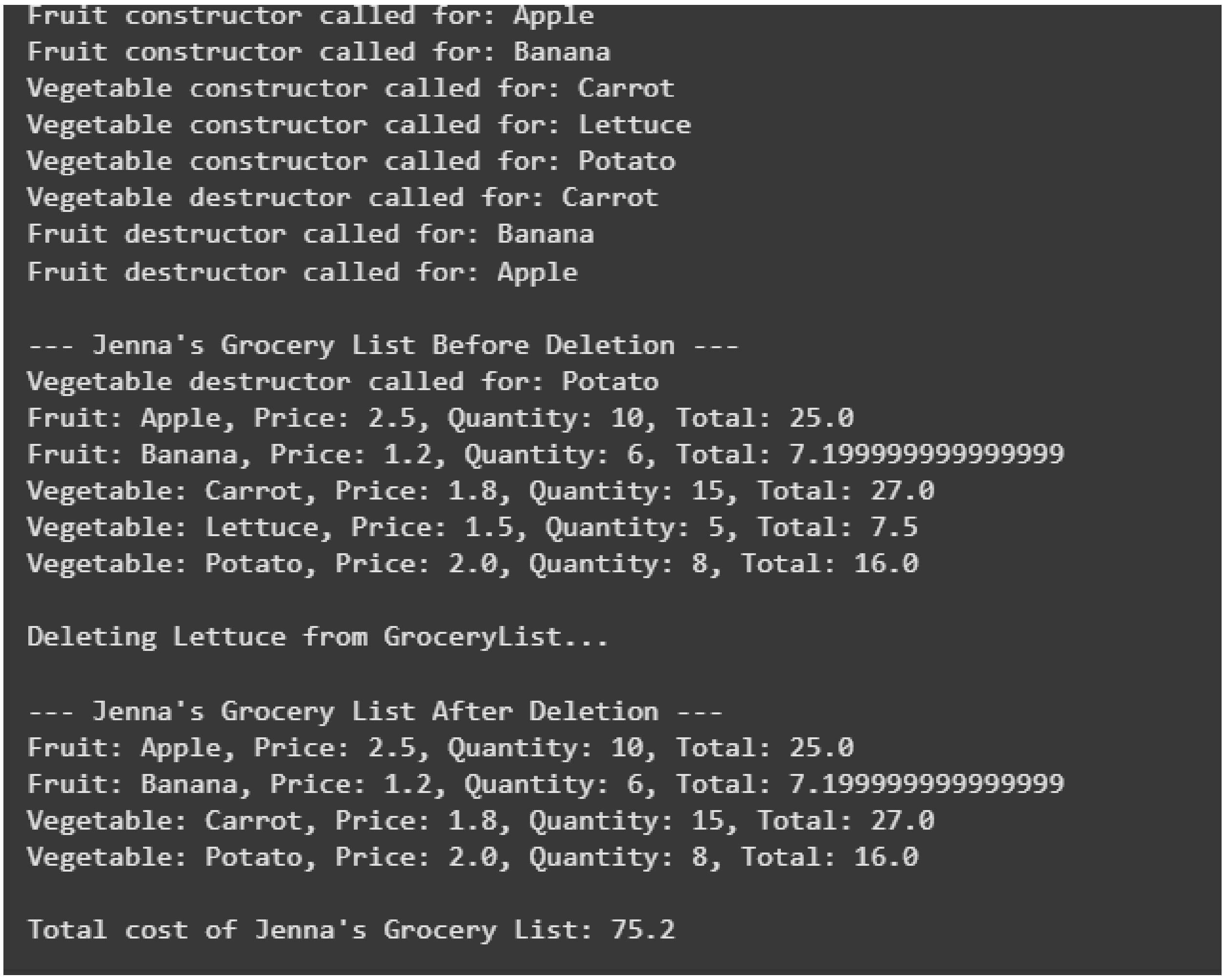


```
Fruit constructor called for: Apple
Fruit constructor called for: Banana
Vegetable constructor called for: Carrot
Vegetable constructor called for: Potato
Vegetable destructor called for: Carrot
Fruit destructor called for: Banana
Fruit destructor called for: Apple

--- Jenna's Grocery List ---
Vegetable destructor called for: Potato
Fruit: Apple, Price: 2.5, Quantity: 10, Total: 25.0
Fruit: Banana, Price: 1.2, Quantity: 6, Total: 7.199999999999999
Vegetable: Carrot, Price: 1.8, Quantity: 15, Total: 27.0
Vegetable: Potato, Price: 2.0, Quantity: 8, Total: 16.0

Total cost of Jenna's Grocery List: 75.2
```

Python uses reference counting and garbage collection**.**

If an object has no remaining reference, its destructor (__del__) is triggered automatically.

In your code, some objects (like Carrot, Banana, Apple) may have been re-created or reassigned, causing the previous instances to be deleted.



```
Fruit constructor called for: Apple
Fruit constructor called for: Banana
Vegetable constructor called for: Carrot
Vegetable constructor called for: Lettuce
Vegetable constructor called for: Potato
Vegetable destructor called for: Carrot
Fruit destructor called for: Banana
Fruit destructor called for: Apple

--- Jenna's Grocery List Before Deletion ---
Vegetable destructor called for: Potato
Fruit: Apple, Price: 2.5, Quantity: 10, Total: 25.0
Fruit: Banana, Price: 1.2, Quantity: 6, Total: 7.199999999999999
Vegetable: Carrot, Price: 1.8, Quantity: 15, Total: 27.0
Vegetable: Lettuce, Price: 1.5, Quantity: 5, Total: 7.5
Vegetable: Potato, Price: 2.0, Quantity: 8, Total: 16.0

Deleting Lettuce from GroceryList...

--- Jenna's Grocery List After Deletion ---
Fruit: Apple, Price: 2.5, Quantity: 10, Total: 25.0
Fruit: Banana, Price: 1.2, Quantity: 6, Total: 7.199999999999999
Vegetable: Carrot, Price: 1.8, Quantity: 15, Total: 27.0
Vegetable: Potato, Price: 2.0, Quantity: 8, Total: 16.0

Total cost of Jenna's Grocery List: 75.2
```

Constructors created the items.

Destructors were called automatically by Python (sometimes earlier than expected, due to garbage collection).

The grocery list was displayed before and after deletion.

The total cost calculation worked correctly.

IV. Conclusion

In conclusion this laboratory demonstrated how constructors are triggered upon object instantiation and how destructors (__del__) are automatically called once an object loses all references or goes out of scope. This was particularly evident when some objects were deleted prematurely before displaying the grocery list, emphasizing the importance of managing references carefully to ensure data persistence. Additionally, the implementation of a TotalSum() function highlighted the ease of iterating over a list of objects to compute their combined value, showcasing object-oriented programming's ability to manage related data efficiently.

References

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