UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Balaoro, Judge Wayne B.

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

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()
      content = ("----------------------------------------------------------------\n")
      content += (f"Fruit: {self.name}, ")
      content += (f"Price: {self.price}, ")
      content += (f"Quantity: {self.quantity}, ")
      content += (f"Subtotal: PHP {subtotal}")
      return content

  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()
      content = ("----------------------------------------------------------------\n")
      content += (f"Vegetable: {self.name}, ")
```

```python
      content += (f"Price: {self.price}, ")
      content += (f"Quantity: {self.quantity}, ")
      content += (f"Subtotal: PHP {subtotal}")
      return content

  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

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:        ")
for item in grocery_list:
    print(item.display())
print("----------------------------------------------------------------")
total = TotalSum(grocery_list)
print(f"Total sum of each grocery: PHP {total}.")


print("\nDestructor called, removing lettuce.\n")
grocery_list.remove(lettuce)
del lettuce

print("         Jenna's Grocery List:        ")
for item in grocery_list:
    print(item.display())
print("----------------------------------------------------------------")

total = TotalSum(grocery_list)
print(f"Total sum of each grocery: PHP {total}.\n")
```

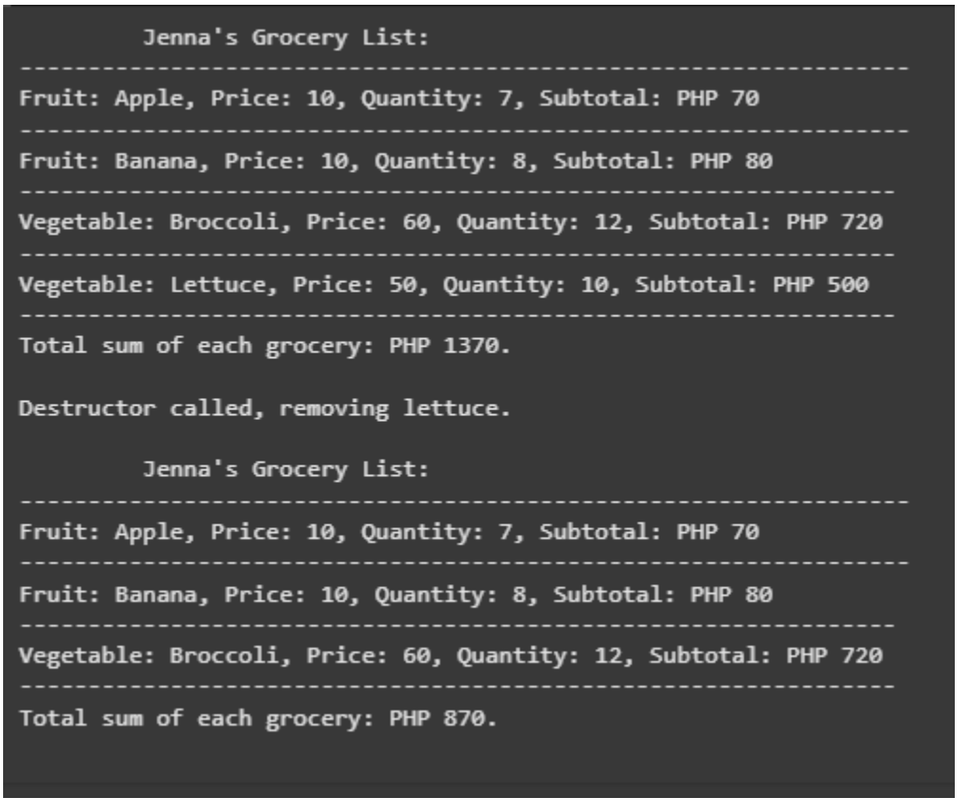

Figure 1: Source Code

This Python source code uses classes and arrays to manage a grocery list. A main class, **GroceryItem**, stores details like name, price, and quantity, while **Fruit** and **Vegetable** inherit these features such as name, price, quantity. The program creates an array of four items (Apple, Banana, Broccoli, Lettuce), displays them, and calculates the total price. It then removes Lettuce from the array, shows the updated list, recalculates the total, and demonstrates copying an item. This mainly shows how to store, update, and remove objects in an array while applying basic object-oriented programming.

IV. Conclusion

This activity showed how arrays can store and manage multiple objects in Python. By adding, displaying, removing, and copying grocery items, we learned how arrays work with object-oriented programming. It also highlighted how arrays make it easy to organize and update related data in a single structure. Through creating a grocery list, we practiced looping through arrays, calculating totals, and modifying elements. Overall, this activity strengthened our understanding of arrays while also introducing useful concepts 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.

[2] GeeksforGeeks. (2025, July 11). *Destructors in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/python/destructors-in-python