Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| LAPUT, MARK DANIELLE E. | Engr. Maria Rizette H. Sayo |

08, 24, 2025

# 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

# Methods

Jenna’s Grocery

A list of grocery items

AI-generated content may be incorrect.

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.

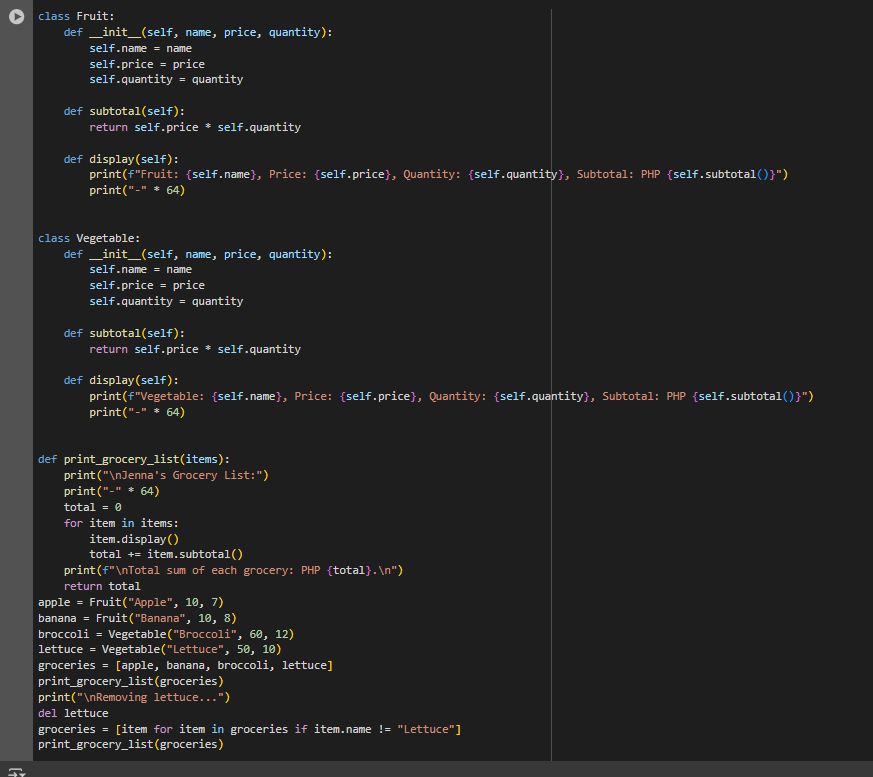
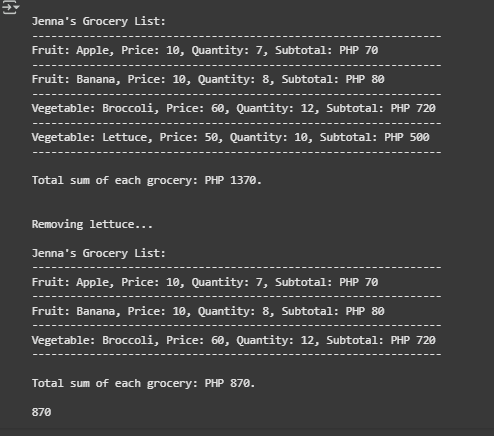
Problem 1: Create a class for the fruit and the vegetable classes. Each class must have a constructor, deconstructor, copy constructor and copy assignment operator. They must also have all relevant attributes (such as name, price and quantity) and functions (such as calculate sum) as presented in the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

# Results



# Conclusion

In this task, we created Python classes for Fruit and Vegetable. Each class has attributes for name, price, and quantity. We included methods to calculate subtotals and show details. We also managed a grocery list by adding and removing items. This demonstrated basic object-oriented programming concepts like constructors and destructors. We made sure to control printed messages for clean output. This exercise reinforced basic OOP principles and practical class management in Python while closely following the problem requirements.

**References**

[1 Real Python — Object-Oriented Programming (OOP) in Python 3