



UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Ruperto, April Anne A.

Instructor:
Engr. Maria Rizette H. Sayo

August 16, 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

Please follow this link: [CPE-201L-DSA-2-A/Laboratory 4/DSA_Lab4.ipynb at main · Ruperto-](#)

```
class Item:
    def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    def __del__(self):
        print(f"{self.name} has been deleted.")

    def __copy__(self):
        return Item(self.name, self.price, self.quantity)

    def __repr__(self):
        return f"{self.name} - Php{self.price}, {self.quantity}x"

    def calculate_total(self):
        return self.price * self.quantity

class Fruit(Item):
    def __init__(self, name, price, quantity):
        super().__init__(name, price, quantity)

    def __repr__(self):
        return f"Fruit: {super().__repr__()}"

class Vegetable(Item):
    def __init__(self, name, price, quantity):
        super().__init__(name, price, quantity)

    def __repr__(self):
        return f"Vegetable: {super().__repr__()}"

def total_sum(grocery_list):
    total = sum(item.calculate_total() for item in grocery_list)
    return total
```

```
def main():
    grocery_list = [
        Fruit("Apple", 10, 7),
        Fruit("Banana", 10, 8),
        Vegetable("Broccoli", 60, 12),
        Vegetable("Lettuce", 50, 10)
    ]

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

    total = total_sum(grocery_list)
    print(f"\nTotal Sum: Php{total}")

    grocery_list = [item for item in grocery_list if item.name != "Lettuce"]

    print("\nUpdated Grocery List after removing Lettuce:")
    for item in grocery_list:
        print(item)

    print("\nClearing the Grocery List")
    del grocery_list

if __name__ == "__main__":
    main()
```

Jenna's Grocery List:
Fruit: Apple - Php10, 7x
Fruit: Banana - Php10, 8x
Vegetable: Broccoli - Php60, 12x
Vegetable: Lettuce - Php50, 10x

Total Sum: Php1370

Updated Grocery List after removing Lettuce:
Lettuce has been deleted.
Fruit: Apple - Php10, 7x
Fruit: Banana - Php10, 8x
Vegetable: Broccoli - Php60, 12x

Clearing the Grocery List
Banana has been deleted.
Apple has been deleted.
Broccoli has been deleted.

[April-Anne/CPE-201L-DSA-2-A](#)

Figure 1 and 2: Screenshot of the Program

IV. Conclusion

The Python program effectively manages Jenna's grocery list by utilizing object-oriented programming principles through the creation of 'Item', 'Fruit', and 'Vegetable' classes. Each class encapsulates relevant attributes and methods, allowing for the calculation of total costs and the distinction between fruits and vegetables. The program initializes a grocery list with various items, displays their details, calculates the total sum, and allows for the removal of specific items, such as 'Lettuce'. By employing a clear structure with classes and methods, the program not only demonstrates functionality but also maintains readability and extensibility, making it a practical example of applying object-oriented concepts to solve real-world problems. Future enhancements could include user input for grocery items or integration with a database for persistent storage.

References

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