UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 4

Arrays

Submitted by:
Uy, Junichiro H.

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

```python
class groceryItem:
  def __init__(self, name, price, quantity):
    self.name = name
    self.price = float(price)
    self.quantity = int(quantity)

  def copy_constructor(self):
    return type(self)(self.name, self.price, self.quantity)

  def copyAssignment(self, new):
    self.name, self.price, self.quantity = new.name, new.price, new.quantity

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

  def details(self):
    return f"{self.quantity} pcs of {self.name} for ₱{self.price} each: ₱{self.sumTotal()} in total!"

  def __del__(self):
    print(f"{self.name} is being deleted!")

class fruits(groceryItem): pass
class vegetables(groceryItem): pass

# Problem 1
apple = fruits("Apple", 10, 7)
banana = fruits("Banana", 10, 8)
broccoli = vegetables("Broccoli", 60, 12)
lettuce = vegetables("Lettuce", 50, 10)
```

Figure 1 Screenshot of program from colab

For problem 1, I made an object called “groceryItem” meaning it is the identity item of any element I will put inside the “groceryList” that contains a name, price, and quantity. This stage lets me reuse the object and create different objects with different parameters and put them under a variable.

```python
# Problem 2
groceryList = [apple, banana, broccoli, lettuce]

print("\nGrocery List:")
for item in groceryList:
  print(item.details())
```

Figure 2 Screenshot of program from colab

Then I put those variables inside a list named groceryList, because I need to use it to display the information and details much faster than printing one by one. Then I use a for loop to print the details simultaneously and automatically.

```python
# Problem 3
def sumAllitems(items):
  totalSum = 0
  for item in items:
    totalSum += item.sumTotal()
  return totalSum

print("Total price before deletion: ", f"₱{sumAllitems(groceryList)}")
```

Figure 3 Screenshot of program from colab

I want to show the sum of all the items in the groceryList while also considering the quantity of each item. So I made a function inside the object groceryItem that multiplies the price and quantity to show the total price per item, and then I made a function outside of that object that lets me add all the total prices for all the items in the groceryList.

```python
#Problem 4
Question_1 = input("\nDo you want to delete an item?(y/n): ")
if Question_1 =="y":
  choice = input("\nWhat item do you want to delete?(name): ")
  index = None

  for i, x in enumerate(groceryList):
    if choice.lower() == x.name.lower():
      index = i
      break

  if index is not None:
    dltItem = groceryList.pop(index)
    del dltItem
  else:
    print(f"\nThe item '{choice}' is not in the list.")

else:
  print("Ok!")

print("\nGrocery List:")
for item in groceryList:
  print(item.details())
```

Figure 4 Screenshot of program from colab

Deleting an object inside a list using pop and using delete instructor is actually hard without using a for loop that lets me read the index and object inside a list, so that's exactly what I did. I used a for loop to read the index in the list and used 'i' to read the index number and then 'x' for the object. Then I used an if function to read if the name of the 'x' or object is equal to the

string provided in ‘choice’; if yes, then I assign a variable that pops the ‘i’ and use that variable to delete the object.

I. Conclusion

Thanks to this library, I learned how to even utilize for loops more. It also gave me a deeper understanding of how to use objects and the instructors, like copy and delete. I think I might have more use for this more in the future. I am looking forward to creating more similar projects in the near future.

References

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