



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 5

---

# Implementation of Arrays

---

*Submitted by:*  
Catahan, Joshua A.

*Instructor:*  
Engr. Maria Rizette H. Sayo

08, 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
- Writing a python program that can implement Array data structure

# II. Methods

- Write a Python program to create an array of 10 integers and display the array items. Access individual elements through indexes and compute for the sum.
- Write a Python program to append a new item to the end of the array. Original array: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Write a Python program to insert a new item before the second element in an existing array. Original array: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Write a Python program to reverse the order of the items in the array. Original array: numbers = [5, 4, 3, 2, 1]
- Write a Python program to get the length of the array. Original array: numbers = [5, 4, 3, 2, 1]

# III. Results

Below are pictures of the codes I have written based on the instructions above:

```
-----
Display array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Sum of all numbers: 55
Index 0: 1
Index 1: 2
Index 2: 3
Index 3: 4
Index 4: 5
Index 5: 6
Index 6: 7
Index 7: 8
Index 8: 9
Index 9: 10
Append a number: 11
With the appended number: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Insert a number: 222
With the inserted number: [1, 222, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
-----
Display array: [1, 2, 3, 4, 5]
Reversing the array
Index 0: 5
Index 1: 4
Index 2: 3
Index 3: 2
Index 4: 1
-----
Display array: [5, 4, 3, 2, 1]
The length of this array is: 5
```

Figure 1: Codes Output

```
numbers = [1,2,3,4,5,6,7,8,9,10] #Original Array
numbers_2 = [1,2,3,4,5] #Original Array
numbers_3 = [5,4,3,2,1] #Original Array

def show_array(self):
    print(f'Display array: {self}')

def calc_sum(self):
    all_numbers = len(self)
    if all_numbers == 0:
        print(self)
    elif all_numbers > 0:
        print(f'Sum of all numbers: {sum(self)}')

def index_array(self):
    for i, element in enumerate(self):
        print(f"Index {i}: {element}")

def append_array(self):
    self.append(int(input(f'Append a number: ')))
    print(f'With the appended number: {self}')

def insert_array(self):
    self.insert(1, int(input(f'Insert a number: ')))
    print(f'With the inserted number: {self}')

def reverse_array(self):
    print("Reversing the array")
    for i, element in enumerate(self[::-1]):
        print(f"Index {i}: {element}")

def show_length(self):
    the_numbers = len(self)
    print(f'The length of this array is: {the_numbers}')

print('-----')
show_array(numbers)
calc_sum(numbers)
index_array(numbers)
append_array(numbers)
insert_array(numbers)
print('-----')
show_array(numbers_2)
reverse_array(numbers_2)
print('-----')
show_array(numbers_3)
show_length(numbers_3)
```

Figure 2: Codes

LINK OF THE CODES IN COLAB:

[https://colab.research.google.com/drive/1sUYYkB6Q1fp3j\\_OW081MTIJhfWCfQsly?authuser=1#scrollTo=uCAOSt37me4Y&line=5&uniqifier=1](https://colab.research.google.com/drive/1sUYYkB6Q1fp3j_OW081MTIJhfWCfQsly?authuser=1#scrollTo=uCAOSt37me4Y&line=5&uniqifier=1)

COMMENTARIES:

For this lab activity, I created a Python script to explore basic array operations like displaying elements, calculating sums, appending and inserting values, reversing arrays, and checking length. I used built-in functions such as `append()`, `insert()`, `sum()`, and `enumerate()` to

perform each task on sample arrays. This helped me understand how arrays respond to different manipulations and how to handle user input effectively.

## IV. Conclusion

In conclusion, this activity reinforced my understanding of array manipulation in Python through practical functions and user interaction. By applying built-in methods and observing their effects, I gained clearer insight into how arrays work and how to structure code for readability and functionality.

## References

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