UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 5

Implementation of 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 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

Please follow this link: CPE-201L-DSA-2-A/Laboratory 5/DSA_Lab5.ipynb at main · Ruperto-April-Anne/CPE-201L-DSA-2-A

SUM

```
[25] def array_sum(arr):
       sum = 0
       for i in arr:
         sum = sum + i
       print("Sum of all Arrays: ", sum)

     arr = [1,2,3,4,5,6,7,8,9,10]
     print("Original Array: ", arr)
     array_sum(arr)
```

```
Original Array:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Sum of all Arrays:  55
```

Figure 1: Screenshot of program

The result shows the original array and displays the sum of all items in the given array.

APPEND

```
[22] def array_append(arr):
       append_arr = int(input("New Element: "))
       arr.append(append_arr)
       print("New Array: ", arr)


     arr = [1,2,3,4,5,6,7,8,9,10]
     print("Original Array: ", arr)
     array_append(arr)
```

```
Original Array:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
New Element: 11
New Array:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

Figure 2: Screenshot of program

The result shows the original array and asks the user for a new element and adding it at the end, then displaying the new array.

INSERT

```
def insert_array(arr):
  insert_arr = int(input("New Element: "))
  position = int(input("Position: "))
  arr.insert(position, insert_arr)
  print("New Array: ", arr)


arr = [1,2,3,4,5,6,7,8,9,10]
print("Original Array: ", arr)
insert_array(arr)
```

```
Original Array:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
New Element: 11
Position: 5
New Array:  [1, 2, 3, 4, 5, 11, 6, 7, 8, 9, 10]
```

Figure 3: Screenshot of program

The result shows the original array and asks the user for a new element and adding it at the specified position, then displaying the new array.

REVERSE

```
def reverse(arr):
  reversed_arr = arr[::-1]
  print("Reversed Array: ", reversed_arr)


arr = [5,4,3,2,1]
print("Original Array: ", arr)
reverse(arr)
```

```
Original Array:  [5, 4, 3, 2, 1]
Reversed Array:  [1, 2, 3, 4, 5]
```

Figure 4: Screenshot of program

The result shows the original array and then reversing the given array, displaying the new array.

LENGHT

```
[19] def array_length(arr):
         count = 0
         for index, element in enumerate(arr):
             count = index + 1
         return count


     arr = [5, 4, 3, 2, 1]
     print("Original Array: ", arr)
     length = array_length(arr)
     print("Length of the array: ", length)
```

```
Original Array:  [5, 4, 3, 2, 1]
Length of the array:  5
```

Figure 5: Screenshot of program

The result shows the original array and displays the number of items in the given array.

IV. Conclusion

To conclusion, this scientific experiment has effectively introduced the basic ideas and methods of the Array data structure. Working with these features in a real-life setting through a Python program has not only our understanding of the array's conception but their practical usage also, which has been further emphasized by their role in problem-solving. This laboratory work has been a positive experience as it has brought about an improvement in our understanding of the concepts and our programming skills, thus equipping us with the knowledge for handling arrays in different programming cases.

References

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