Data Structure and Algorithm

Laboratory Activity No. 5

Implementation of Arrays

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Hermosura, Leigh B. | Engr. Maria Rizette H. Sayo |

August 16, 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
* Writing a python program that can implement Array data structure

# 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]

# Results

The results yielded by this laboratory activity are presented in this chapter. Explanations with their respective visual aids are provided below.

An array is a linear data structure that contains elements of the same types, arranged sequentially, stored in the same block of memory. Arrays have a fixed size, meaning that once the size is specified, it cannot be shrunk to remove gaps or expanded to accommodate new elements [1].

The first method of the activity requires a Python program that displays the sum of an array containing 10 integers. The image of the program is provided below.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 1 Computing the sum of an array

The program initializes two variables: an *array* list containing numbers from 1 to 10 and the *total* variable which is set to 0. The second part prints the *array* and its contents. Lastly, the program runs a for loop that iterates through the elements in *array*, gets their sum and stores the result in *total*, then prints the sum of all the elements.

A computer screen shot of a program code

AI-generated content may be incorrect.

Figure 2 Appending a new item in the array

A screenshot of a computer program

AI-generated content may be incorrect. The second program begins by printing the *numbers* array and requests an input from the user to append to it. It then prints the updated *numbers* list after each input. A for loop is used to rerun the program three times, allowing the user to input multiple numbers into the array.

Figure 3 Inserting before the second element in the array

The process behind the third method is similar to the second method wherein the program prompts the user for a value to be inserted to the *numbers* array. The only difference is that the program uses the *.insert()* method in which we can specify the location where the new value will be inserted into.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 4 Reversing the elements of an array

The fourth method defines the *numbers* array containing elements [5, 4, 3, 2, 1] and prints it. The built in *.reverse()* function is used for the array. When printed, produces the list [1, 2, 3, 4, 5].

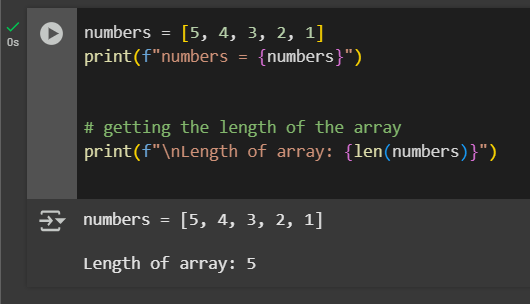


Figure 5 Printing the length of an array

The fifth method defines an array called *numbers* that has elements [5, 4, 3, 2, 1] and displays the list. The *len()* function is then utilized to determine the length of the array which is then printed, showing a value of 5.

# Conclusion

This laboratory activity effectively demonstrated fundamental array operations in Python, such as summation, insertion, appending, reversing, and determining the length of an array. Through hands-on implementation, the behavior and structure of arrays were explored in detail. Each method highlighted a specific functionality, reinforcing the concept of arrays as fixed-size, sequential data structures. The activity not only deepened the understanding of how arrays work in memory but also provided practical experience with common array manipulation techniques.

**References**

[1] GeeksforGeeks, “What is Array?,” *GeeksforGeeks*, Oct. 27, 2017. https://www.geeksforgeeks.org/dsa/what -is-array/

‌