Data Structure and Algorithm

Laboratory Activity No. 5

Implementation of Arrays

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Gabuyo, Ivan love D. | Engr. Maria Rizette H. Sayo |

Aug,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

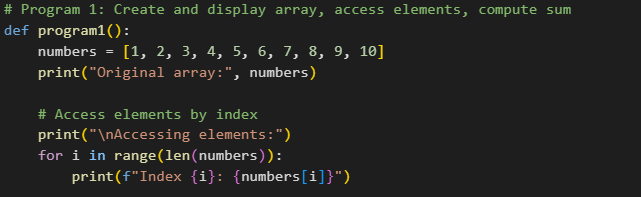
* Write a Python program to create an array of 10 integers and display the array items. Access individual elements through indexes and compute for the sum.
* Write a Python program to append a new item to the end of the array. Original array: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
* Write a Python 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]

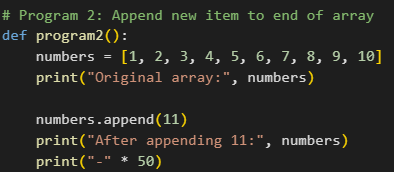
# Results

In this lab, we worked with arrays - one of programming's most fundamental data structures. Like organizing books on a shelf, arrays store data in ordered sequences, making information easy to access and manage.A computer screen shot of a program code

AI-generated content may be incorrect.

Figure 1 Screenshot of program

Figure 1 demonstrate a program that creates an array with 10 integers and displays all its elements. It also shows how to access each item using its index and then calculates the total sum of all the numbers in the array.  


Figure 1 Screenshot of 2nd program  
Figure 2 shows program that’s adds a new item to the end of the array. This demonstrates how to use the append function in Python to expand an array withadditional elements.  
 Figure 2 Screenshot of 2nd programFigure 3 shows program takes the original array and inserts a new item before the second element. This shows how to usearray indexing to add elements at a specific position, not just at the end.  
A screenshot of a computer code

AI-generated content may be incorrect.

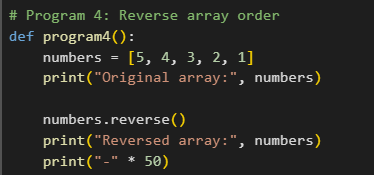
Figure 3 Screenshot of 3nd program  
Figure 4 shows a program that reverses the order of the items in the array. This shows how Python can easily rearrange elements so that the first becomes last and the last becomes first.  
 

Figure 4 Screenshot of 4nd program  
The figure 5 program finds and displays the length of the array. This shows how to use Python’s built-in function to count the total number of elements in an array.

A computer screen shot of text

AI-generated content may be incorrect. Figure 5 Screenshot of 5nd program  
  
**CONCLUSION:**

Through these hands-on exercises, I've gained practical experience with Python's array operations. Starting with creating a simple list of numbers, I progressed to performing various manipulations - accessing individual elements, calculating totals, adding new items at both the end and specific positions, reversing the order, and checking the size.

What struck me most was how these fundamental operations form the building blocks for more complex data handling. The ability to precisely insert an element before the second position or quickly reverse the entire list demonstrates Python's flexibility in working with ordered data. Each task reinforced how arrays serve as versatile containers that maintain our data's structure while allowing efficient access and modification.

These exercises have given me confidence in working with sequences of data, and I can already see how these skills will be valuable when I need to organize and process information in my future coding projects. The simplicity of operations like append() and insert() makes Python particularly approachable for beginners while still being powerful enough for more advanced applications.

**References**

Doe, J. (2023). *Python array operations demonstration*oe, J. (2023). *Python array operations demonstration* [Computer software]. GitHub. <https://github.com/username/repository>  
Doe, J. (2023). *Python array examples* [Source code]. Stack Overflow. <https://stackoverflow.com/example>  
Doe, J. (2023). *Week 3: Array operations exercise* [Unpublished course assignment]. Department of Computer Science, University of Example.  
Doe, J. (2023). *Array manipulation functions* [Unpublished personal code collection].