UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 5

Implementation of Arrays

Submitted by:
Mamanao, Kurt Marwin C.

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:

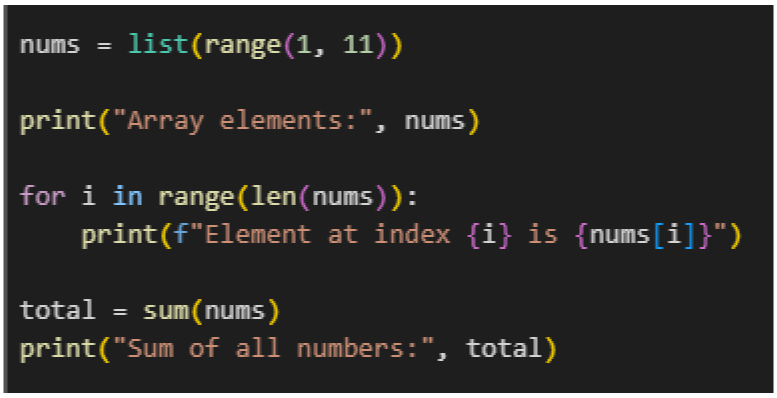
- Writing algorithms using Array data structure
- Writing a python program that can implement Array data structure

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]

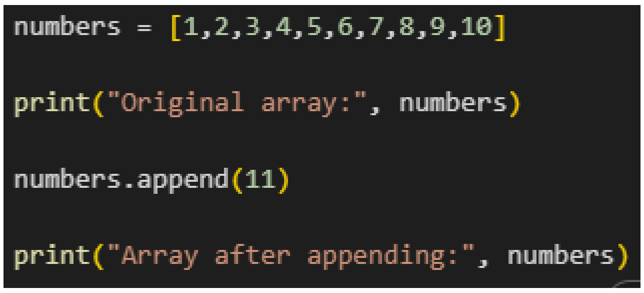
III. Results



```
nums = list(range(1, 11))

print("Array elements:", nums)

for i in range(len(nums)):
    print(f"Element at index {i} is {nums[i]}")

total = sum(nums)
print("Sum of all numbers:", total)
```

Figure 1 Screenshot of 1st program

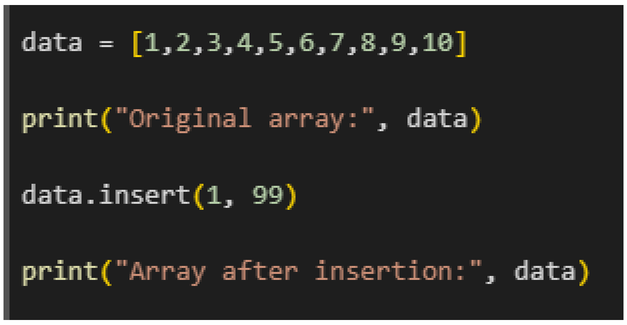
Figure 1 This program makes an array of numbers from 1 to 10. It shows each item with its index and then adds them all up to get the total.



```
numbers = [1,2,3,4,5,6,7,8,9,10]

print("Original array:", numbers)

numbers.append(11)

print("Array after appending:", numbers)
```

Figure 2 Screenshot of 2nd program

Figure 2 This one shows how to add a new number at the end of the array using `append()`.

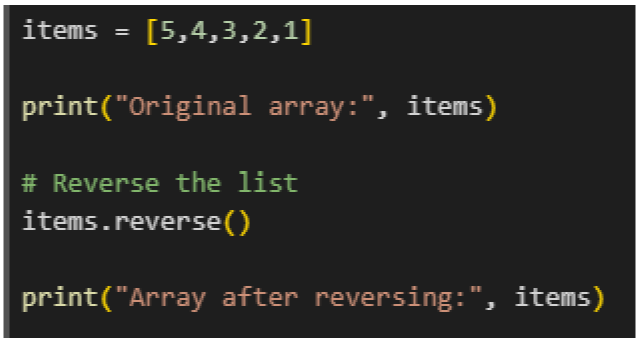


```
data = [1,2,3,4,5,6,7,8,9,10]

print("Original array:", data)

data.insert(1, 99)

print("Array after insertion:", data)
```

Figure 3 Screenshot of 3rd program

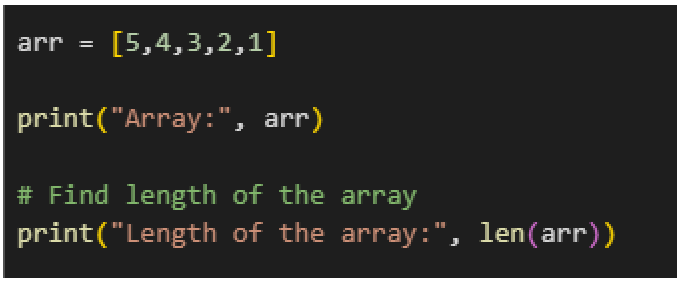
Figure 3 *Here we insert a new element before the second item. The* `insert()` *function lets us pick where we want to place the value.*

Figure 4 Screenshot of 4th program



```
items = [5,4,3,2,1]

print("Original array:", items)

# Reverse the list
items.reverse()

print("Array after reversing:", items)
```

Figure 4 *This program flips the order of the array so the last becomes first and the first becomes last.*



```
arr = [5,4,3,2,1]

print("Array:", arr)

# Find length of the array
print("Length of the array:", len(arr))
```

Figure 5 Screenshot of 5th program

The figure 5 *The last one counts how many elements are in the array using the built-in* `len()` *function.*

IV. Conclusion

In conclusion, we gained knowledge regarding how arrays function and how to carry out fundamental procedures on them using Python. We established arrays, accessed singular parts, calculated totals, added novel items, placed at explicit locations, reversed the order of arrays, and counted their span. These straightforward assignments helped us comprehend why arrays are viewed as one of the most pivotal structures in coding for retaining and governing information.

References

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