St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology
A.Y. 2023-24
Class: SE-ITA/B, Semester: IV
Subject: **Python Lab**.

Experiment – 5: Python Data Structures-I

1. **Aim:** To implement a Python program to demonstrate use of Python arrays

2. **Objectives:** After performing this experiment, the student will be able to understand and use arrays in Python

3. **Outcomes:** Students shall be able to **apply** various data structures and functions in Python. (LO-404.2).

4. **Prerequisite:** Knowledge of Python data types, Basics of data structures.

5. **Requirements:** Personal Computer (PC), Windows /Linux Operating System, Python IDE.

6. **Pre-Experiment Exercise:**

Theory:

An array is an object that stores a group of elements of same data type. Arrays can increase or decrease their size dynamically. In Python, there is a standard module by the name 'array' that helps in creating and using arrays.

There are three ways of importing arrays into a Python program.

Importing entire array module and using array class to create an array.

import array as arr

a = array.array ('i', [4,6,2,9])

Importing array module by giving it an alias name.

Import array as ar

a = ar.array('*i*', [4,6,2,9])

Importing array module by using '*'

*from array import **

a = array ('i',[4,6,2,9])

When creating an array type code must be specified as:

arrayname = array(type code, [elements])

The type code '*i*' represents integer type array while type code '*f*' represents float type array *etc*.

An index represents the position number of an element in an array. There are several ways in which an array can be sliced using indexing.

The arrays class of module array offers various methods which can be used for simple processing of arrays.

7. Laboratory Exercise

A. Procedure

i. Open IDLE IDE for Python programming
ii. Open new Python file from menu file-new
iii. Type Python code with proper syntax
iv. Save file with .py extension
v. Execute the command statements inside the saved file using cntr+enter key and explore results in other windows of IDLE.

B. Program code with comments:

Write and execute your program code to achieve the given aim and attach it with your own comments with neat indentation.

8. Post-Experiments Exercise

A. Extended Theory:

1. What is numpy? How arrays can be careated using Numpy?
2. List down statistical operations that can be performed on arrays.
3. Write Python statements used to create an array whose all elements are zeros.

B. Results/Observations/Program output:

Present the program input/output results and comment on the same.

C. Questions/Programs:

1. Write a Python program to create an integer array of four elements.
2. Write a Python program to create an array of *double* data type. Using this array, create another array whose elements are three times that of the elements of the first array.
3. Write a Python program to understand slicing and indexing of arrays.
4. Write a Python program to understand various methods of array class.
5. Write a Python program to implement a stack using an array.

D. Conclusion:

1. Write what was performed in the experiment/program.
2. What is the significance of experiment/program?
3. Mention a few applications of what was studied.

E. References

1. Dr. R. Nageswara Rao," Core Python Programming", Dreamtech Press, Wiley Publication
2. Zed A. Shaw "LEARN PYTHON THE HARD WAY" Addison Wesley

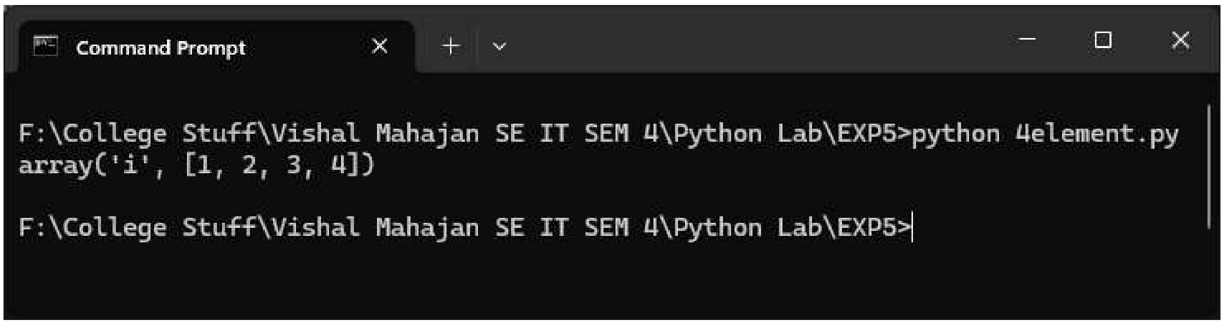
3. What is NumPy? — NumPy v1.20 Manual

1. Write a Python program to create an integer array of four elements

Code:

```
import array


arr = array.array('i', [1, 2, 3, 4])
print(arr)
```

Output:



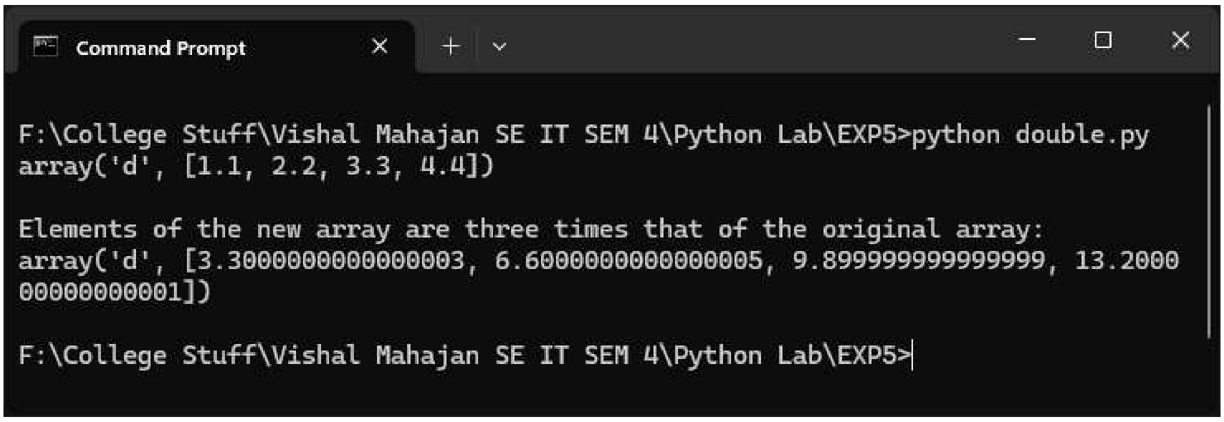
```
F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>python 4element.py
array('i', [1, 2, 3, 4])

F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>
```

2.Write a Python program to create an array of *double* data type. Using this array, create another array whose elements are three times that of the elements of the first array.

Code:

```
import array


arr = array.array('d', [1.1, 2.2, 3.3, 4.4])
print(arr)


# Using list comprehension
arr2 = array.array('d', [i*3 for i in arr])
print("\nElements of the new array are three times that of the
original array:")
print(arr2)
```

Output:



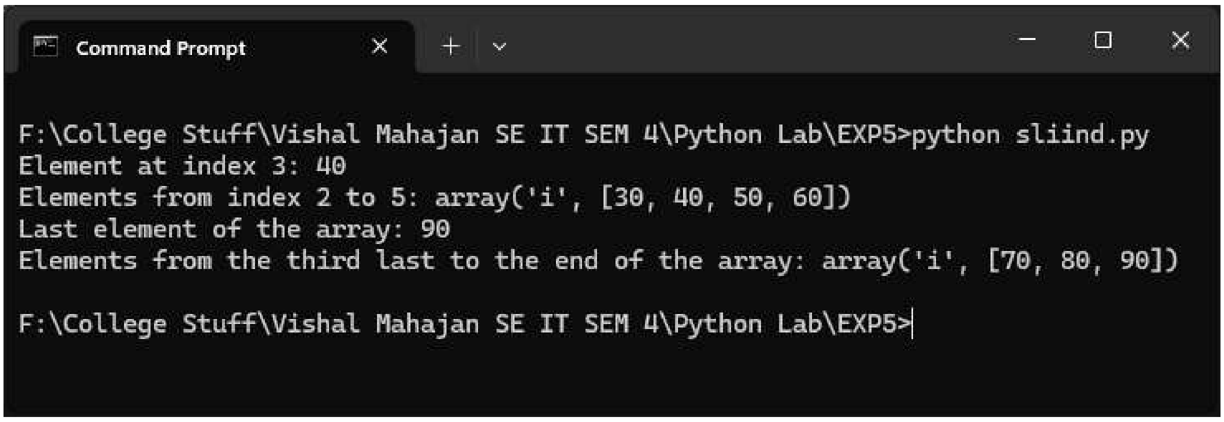
```
F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>python double.py
array('d', [1.1, 2.2, 3.3, 4.4])

Elements of the new array are three times that of the original array:
array('d', [3.3000000000000003, 6.6000000000000005, 9.899999999999999, 13.2000
00000000001])

F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>
```

3. Write a Python program to understand slicing and indexing of arrays

Code:

```
# Create an array
arr = array.array('i', [10, 20, 30, 40, 50, 60, 70, 80, 90])

# Indexing: Accessing an element of the array using its index
print("Element at index 3:", arr[3])

# Slicing: Getting a subset of the array using a range of indices
print("Elements from index 2 to 5:", arr[2:6])

# Negative indexing: Accessing elements from the end of the array
print("Last element of the array:", arr[-1])

# Negative slicing: Getting a subset from the end of the array
print("Elements from the third last to the end of the array:",
arr[-3:])
```

Output:



```
F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>python sliind.py
Element at index 3: 40
Elements from index 2 to 5: array('i', [30, 40, 50, 60])
Last element of the array: 90
Elements from the third last to the end of the array: array('i', [70, 80, 90])

F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>
```

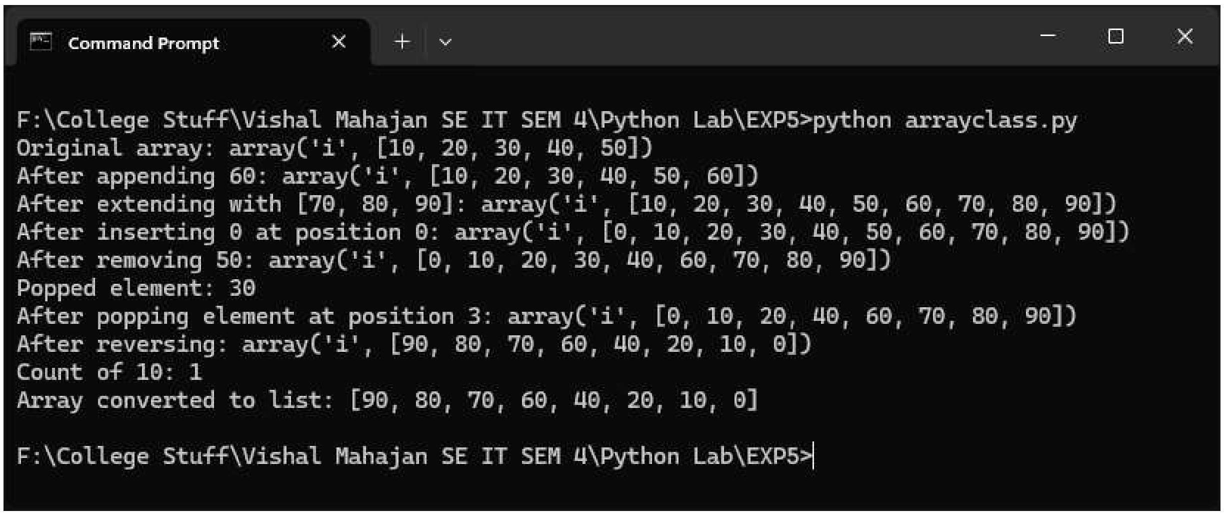
4.Write a Python program to understand various methods of array class.

Code:

```
import array

# Create an array
arr = array.array('i', [10, 20, 30, 40, 50])

print("Original array:", arr)

# append(): Adds an element to the end of the array
arr.append(60)
print("After appending 60:", arr)

# extend(): Adds multiple elements to the end of the array
arr.extend([70, 80, 90])
print("After extending with [70, 80, 90]:", arr)

# insert(): Inserts an element at a specific position
arr.insert(0, 0)
print("After inserting 0 at position 0:", arr)

# remove(): Removes the first occurrence of an element
arr.remove(50)
```

```python
print("After removing 50:", arr)

# pop(): Removes and returns the element at a specific position
print("Popped element:", arr.pop(3))
print("After popping element at position 3:", arr)

# reverse(): Reverses the array
arr.reverse()
print("After reversing:", arr)

# count(): Returns the number of occurrences of an element
print("Count of 10:", arr.count(10))

# tolist(): Converts the array to a list
print("Array converted to list:", arr.tolist())
```

Output:



```
F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>python arrayclass.py
Original array: array('i', [10, 20, 30, 40, 50])
After appending 60: array('i', [10, 20, 30, 40, 50, 60])
After extending with [70, 80, 90]: array('i', [10, 20, 30, 40, 50, 60, 70, 80, 90])
After inserting 0 at position 0: array('i', [0, 10, 20, 30, 40, 50, 60, 70, 80, 90])
After removing 50: array('i', [0, 10, 20, 30, 40, 60, 70, 80, 90])
Popped element: 30
After popping element at position 3: array('i', [0, 10, 20, 40, 60, 70, 80, 90])
After reversing: array('i', [90, 80, 70, 60, 40, 20, 10, 0])
Count of 10: 1
Array converted to list: [90, 80, 70, 60, 40, 20, 10, 0]

F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>
```

5.Write a Python program to implement a stack using an array

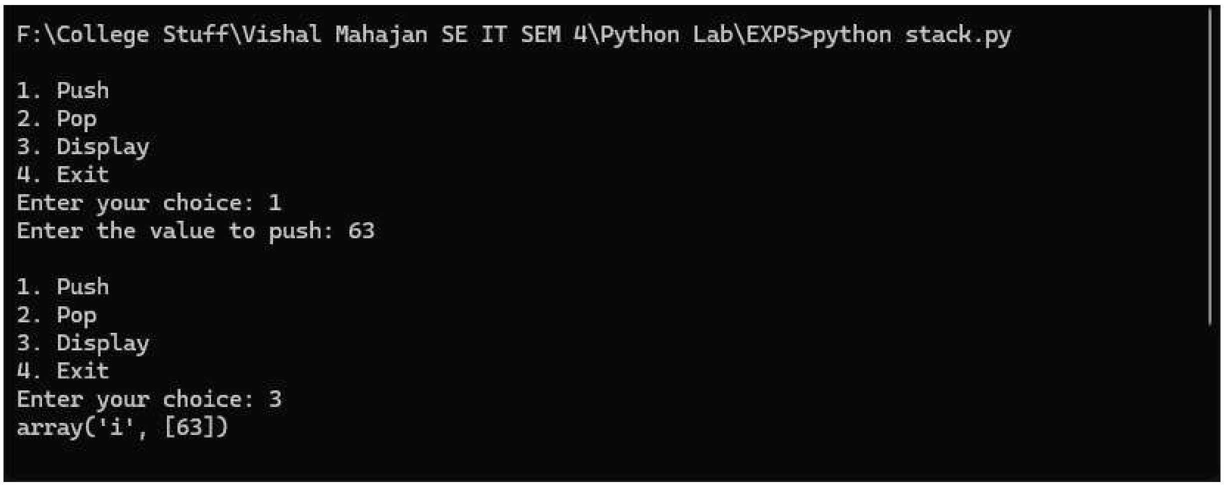
Code:

```python
import array


# Initialize an empty array as a stack
```

```python
stack = array.array('i')

# Push operation
def push(val):
    stack.append(val)

# Pop operation
def pop():
    if len(stack) == 0:
        print("Stack is empty")
    else:
        return stack.pop()

# Display stack
def display():
    print(stack)



while(True):
    print("\n1. Push\n2. Pop\n3. Display\n4. Exit")
    choice = int(input("Enter your choice: "))

    if choice == 1:
        val = int(input("Enter the value to push: "))
        push(val)
    elif choice == 2:
        popped = pop()
        if popped is not None:
            print(f"Popped element: {popped}")
    elif choice == 3:
        display()
    elif choice == 4:
        break
    else:
        print("Invalid choice. Try again.")
```

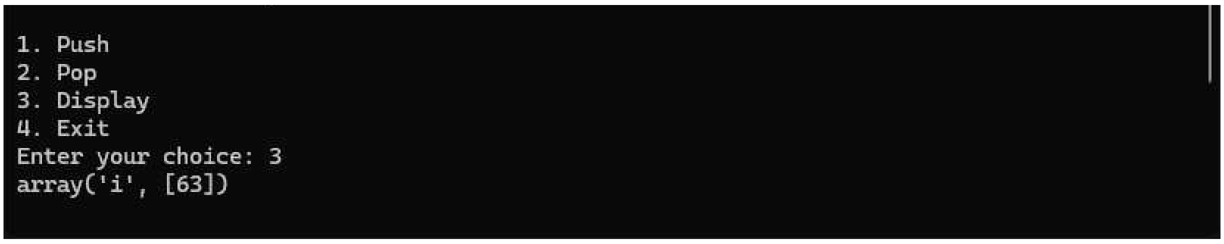
Output:

1.Push



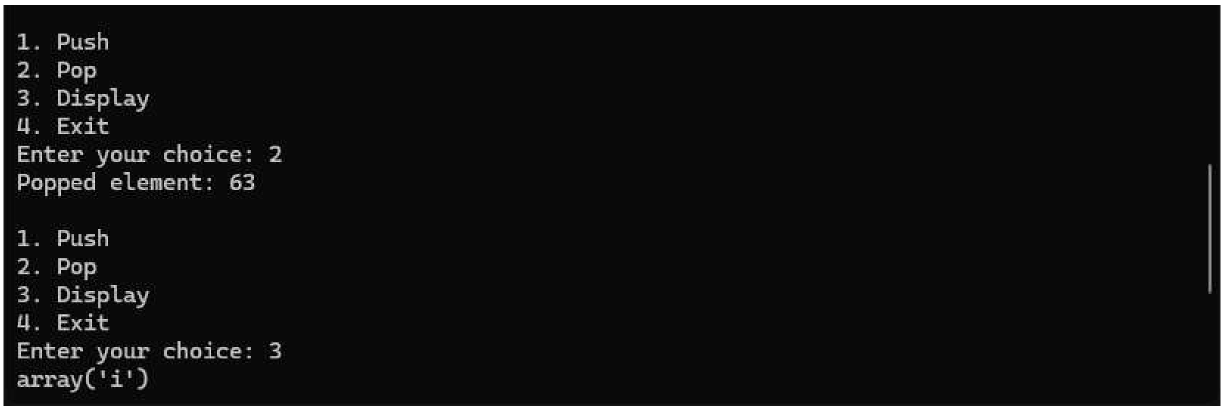
```
F:\College Stuff\Vishal Mahajan SE IT SEM 4\Python Lab\EXP5>python stack.py

1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 63

1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
array('i', [63])
```

2.Display



```
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
array('i', [63])
```

3.Pop



```
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Popped element: 63

1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
array('i')
```