

**National University of Computer and Emerging Sciences**



**Laboratory Manual**

*for*

**Object Oriented Programing Lab**

Course Instructor	Mr. Uzair Naqvi
Lab Instructor(s)	Aqib Zeeshan, Seemab Ayub
Section	BCS 2E
Date	Wednesday, 14 Feb 2024
Semester	Spring 2024

**Department of Computer Science**

FAST-NU, Lahore, Pakistan

**Objectives:**

In this lab, students will practice:

1. Pointers
2. Dynamic memory allocation
3. Functions

**Important Note:**

- **There shouldn't be any memory leakage or dangling pointers in your program.**
- Make separate functions for input and output of arrays. Your main should be a sequence of function calls only
- You are not allowed to use global variables and goto instruction
- **Submit only one cpp file having main function testing all the following functions**

**Question 1:****Exercise 1 [Input Array]:**

Write a function **int\* InputArray(int& size)** that asks user to enter size of required array, allocates the memory on heap, takes input in array and returns its pointer.

**Exercise 2 [Output Array]:**

Write a program **void OutputArray(int\* myArray, const int& size)** that takes a pointer to an integer array and prints its data.

Write main function to test above functionality.

**Question 2:****Exercise 3 – Expand Array**

Write a program that keeps taking integer input from the user until user enters -1 and displays the data in reverse order.

Your program should save the input in a dynamically allocated array. Initially create a dynamic array of five integers. Each time the array gets filled your program should double the size of array (i.e. create a new array of double size, copy previous data in new array, delete previous array) and continue taking the input. After receiving -1 (i.e. end of data input) your program should print the numbers in the reverse order as entered by the user.

Note: Write a separate function that **AllocateAndCopyArray** to grow and copy the array. Use **OutputArray** function to print the final array.

**Important Note: subscript operator [] is not allowed to traverse the array. Use only offset notation. i.e instead of using myArray[i] use \*(myArray+i) to read/write an element. Do not consume extra space. There shouldn't be any memory leakage or dangling pointers in your code.**

**Question 3:**

Take size input from the user and create an array of that size. Now populate the array as well by taking input from the user.

- First Implement **void copyArray(int\* arr, int \*&arr1, int size)** that copies arr into arr1.
- Now implement another function **int reduceArray(int \*arr, int \*&arr1, int size)** that asks user to enter size to reduce the array. To reduce the array remove the elements of the arr from the start and copy remaining into arr1. Use **copyArray** function to copy.

**For Example:****Input:**

Please enter size: 8

Please enter elements: 91

5

3

40

7

8

12

642

Please enter the reduced size of array: 5

**Output:**

Array after reduction is: 40

7

8

12

642

**Question 4:**

Fibonacci sequence is a sequence in which every number after the first two is the sum of the two preceding ones. Write a C++ program that takes a number **n** from user and populate a dynamic array with first n Fibonacci numbers.

**For example:**

For n=10

Fibonacci Numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55