**National University of Computer and Emerging Sciences**



**Laboratory Manual**

*for*

**Object Oriented Programing Lab**

| Course Instructor | Mr. Uzair Naqvi |
| --- | --- |
| Lab Instructor(s) | M Hashir, Seemab Ayub |
| Section | BCS-2B |
| Date | Tuesday, 6 Feb 2024 |
| Semester | Spring 2024 |

**Department of Computer Science**

FAST-NU, Lahore, Pakistan

**Objectives:**

In this lab, students will practice:

1. Functions with Pointers
2. Dynamic arrays

**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**

**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.

**Exercise – 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.

**Exercise – Dynamic Array Sum:**

Write a function that takes a dynamic array of integers and its size as arguments. The function should calculate and return the sum of all elements in the array.

Example output:

| Enter the size of the array: 5  Enter 5 elements:  1 2 3 4 5  Sum of array elements: 15 |
| --- |

**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.**