National University of Computer and Emerging Sciences

Lab Manual 02
Object Oriented Programming

Course Instructor	Mr. Usama Hassan
Lab Instructor (s)	Ms. Fariha Maqbool Mr. Sohaib Ahmad
Section	BSE-2C
Semester	Spring 2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Have more understanding of pointers.
- ✓ Access and modify arrays via pointers.
- ✓ Using array pointers for different problems.

Note:

- Create separate functions to perform each task.
- Create only pointers and size variables in main function and call the created functions through the driver function (main function).

Task: 1

Write a C++ program and implement the following functions:

- `void initializeArray(int*& Arr, int& size);// ask the user to enter size of the array and then populate the array.`
- `void growArray(int*& Arr, int& size); // grow the passed array by size one: create a new temporary array of size (size+1), copy all the element of old array to new array, delete the old array, and point the new array to old array.`
- `int* shrinkArray(int* Arr, int& size); // shrink the array by size one.`
- `void printArray(const int* arr, const int size); // print the size and elements of the array.`
- `int main(); // Driver function: create a menu based program that ask the user to perform the above tasks.`

Task: 2

Take two arrays of integers A and B of sizes M and N respectively (M and N taken from User). Then you need to mix these arrays into a third array named C such that the following sequence is followed.

All even numbers of A from left to right are copied into C from left to right.
All odd numbers of A from left to right are copied into C from right to left.
All even numbers of B from left to right are copied into C from left to right.
All old numbers of B from left to right are copied into C from right to left.

A, B and C are the arrays to Mix. e.g., A is {3, 2, 1, 7, 6, 3} and B is {9, 3, 5, 6, 2, 8, 10} the resultant array C is {2, 6, 6, 2, 8, 10, 5, 3, 9, 3, 7, 1, 3}
Display all of the arrays with proper label to show the result.

Task: 3

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.

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.

Task: 4

Write a C++ program that:

- 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.
- `void copyArray(int* arr, int*& arr1, int size);` // that copies arr into arr1.
- `int reduceArray(int* arr, int*& arr1, int size);` // asks user to enter size to reduce the array.
- `int printArray(int* arr, int size);` // this function will print the size and elements of the array, if the array is empty it will say "Array is empty.".
- `int main(); // write a driver function that initialize a variable size and dynamic array, call each function one by one and print the array to show the output on console.`

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
```