**National University of Computer and Emerging Sciences**

****

Lab Manual # 10

Programming Fundamentals

(Section BCS-1J)

|  |  |
| --- | --- |
| Course Instructor | Mr,Owais Idrees |
| Lab Instructor(s) | Abiha Aftab  Muntaha Zaigham |
| Section | BCS-1J |
| Semester | Fall 2021 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Objectives**

The objectives of this lab are to cover the following:

* all operations on array
* Cstrings

**Arrays**:

In C++, an array is a variable that can store multiple values of the same type. For example,

Suppose a class has 27 students, and we need to store the grades of all of them. Instead of creating 27 separate variables, we can simply create an array:

double grade[27];

Here, grade is an array that can hold a maximum of 27 elements of double type.

In C++, the size and type of arrays cannot be changed after its declaration.

## C++ Array Declaration

dataType arrayName[arraySize];

For example,

int x[6];

Here,

* int - type of element to be stored
* x - name of the array
* 6 - size of the array

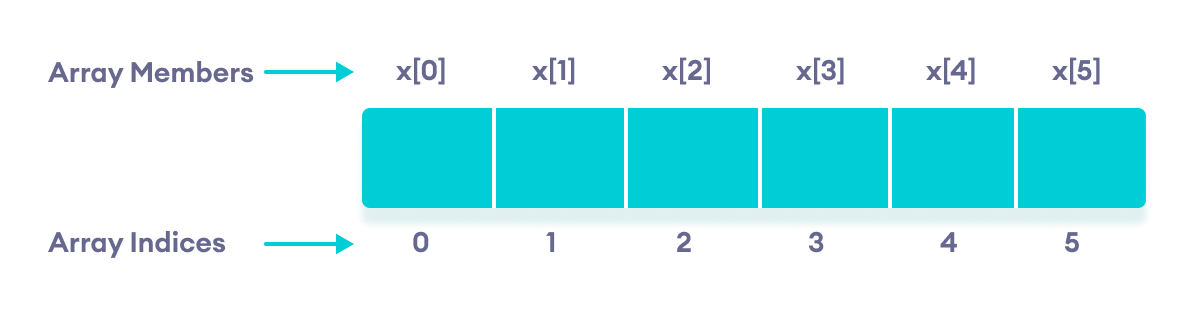
## Access Elements in C++ Array

In C++, each element in an array is associated with a number. The number is known as an array index. We can access elements of an array by using those indices.

// syntax to access array elements

array[index];

Consider the array x we have seen above.

Elements of an array in C++

**CStrings**

String is a collection of characters. There are two types of strings commonly used in C++ programming language:

* Strings that are objects of string class (The Standard C++ Library string class)
* C-strings (C-style Strings)

## C-strings

In C programming, the collection of characters is stored in the form of arrays. This is also supported in C++ programming. Hence it's called C-strings.

C-strings are arrays of type char terminated with null character, that is, \0 (ASCII value of null character is 0).

### How to define a C-string?

char str[] = "C++";

In the above code, **str is a string and it holds 4 characters.**

**Although, "C++" has 3 character, the null character \0 is added to the end of the string automatically.**

### Alternative ways of defining a string

char str[4] = "C++";

char str[] = {'C','+','+','\0'};

char str[4] = {'C','+','+','\0'};

Like arrays, it is not necessary to use all the space allocated for the string. For example:

char str[100] = "C++";

### Example 1: C++ String to read a word

**C++ program to display a string entered by user.**

#include <iostream>

using namespace std;

int main()

{

char str[100];

cout << "Enter a string: ";

cin >> str;

cout << "You entered: " << str << endl;

cout << "\nEnter another string: ";

cin >> str;

cout << "You entered: "<<str<<endl;

return 0;

}

**Output**

Enter a string: C++

You entered: C++

Enter another string: Programming is fun.

You entered: Programming

Notice that, in the second example only "Programming" is displayed instead of "Programming is fun".

This is because the extraction operator >> works as scanf() in C and considers a space " " has a terminating character.

### Example 2: C++ String to read a line of text

**C++ program to read and display an entire line entered by user.**

#include <iostream>

using namespace std;

int main()

{

char str[100];

cout << "Enter a string: ";

cin.get(str, 100);

cout << "You entered: " << str << endl;

return 0;

}

**Output**

Enter a string: Programming is fun.

You entered: Programming is fun.

To read the text containing blank space, cin.get function can be used. This function takes two arguments.

First argument is the name of the string (address of first element of string) and second argument is the maximum size of the array.

In the above program, str is the name of the string and 100 is the maximum size of the array.

# Lab Manual:

**Problem: 1 (Arrays)**

Create a function that takes an array of numbers arr, a char s and return an array of numbers as per the following rules:

* "A" returns a sorted array in ascending order.
* "D" returns a sorted array in descending order.
* "N" returns an array without any modification.

### Examples

AscDesNone([4, 3, 2, 1], "A" ) ➞ [1, 2, 3, 4]

AscDesNone([7, 8, 11, 66], "D") ➞ [66, 11, 8, 7]

AscDesNone([1, 2, 3, 4], "N") ➞ [1, 2, 3, 4]

#### **Problem: 2 (C-strings)**

Write a function that returns the frequency of a character in string. Use C-strings.

##### Example:

Enter the string: I love programming

Enter character to find: o

‘o’ occurred 2 time in your entered string.

**Problem: 3 (C-Strings)**

Write a function to Remove all Characters including spaces in a String Except Alphabets.

Example:

Enter a string: I2love !Programming

Filtered String: IloveProgramming

**Problem: 4 (Arrays and C-Strings)**

Create a function that tweaks letters by one forward (+1) or backwards (-1) according to an array and store the result in an array named tweakedArray and print it in the function.

### Examples

tweakLetters("apple", {0, 1, -1, 0, -1}) ➞ "aqold"

// "p" + 1 => "q"; "p" - 1 => "o"; "e" - 1 => "d"

tweakLetters("many", {0, 0, 0, -1}) ➞ "manx"

tweakLetters("rhino", {1, 1, 1, 1, 1}) ➞ "sijop"

### Notes

Don't worry about capital letters.

**Problem: 5 (Arrays and C-Strings)**

Create two functions toCamelCase() and toSnakeCase() that each take a single string and convert it into either camelCase or snake\_case.

### Examples

toCamelCase("hello\_edabit") ➞ "helloEdabit"

toSnakeCase("helloEdabit") ➞ "hello\_edabit"

toCamelCase("is\_modal\_open") ➞ "isModalOpen"

toSnakeCase("getColor") ➞ "get\_color"