**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Object Oriented Programming Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 3rd**

**Submission Date: 08 February, 2022**

**Submitted to: Engr. Rafi-Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 03**

**Strings in C++**

**Objectives:**

In this lab we will be discussing Strings in detail. This is one of the most important concepts in C++ language. A string is a sequence of characters. Any sequence or set of characters defined within double quotation symbols is a constant string. In C++ it is required to do some meaningful operations on strings they are:

* Reading string
* Displaying strings
* Combining or concatenating strings
* Copying one string to another.
* Comparing string & checking whether they are equal
* Extraction of a portion of a string

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

**Arrays of Characters**

**Streaming an Array of Characters**

* Like any other variable, before using a string, you must first declare it, which is done by type the char keyword, followed by the name of the variable, followed by square brackets. When declaring the variable, if/since you do not know the number of characters needed for the string; you must still provide an estimate number. You can provide a value large enough to accommodate the maximum number of characters that would be necessary for the variable. For a person's name, this could be 20. For the title of a book or a web page, this could be longer.
* Here are examples:
* char Name[20];
* char BookTitle[40];
* char WebReference[80];
* char WeekDay[4];

**Multidimensional Arrays of Characters**

* C/C++ treats arrays of characters differently than it does the other arrays. For example, we have learned to declare a two-dimensional array of integers as follows:
* int Number[2][6] = { { 31, 28, 31, 30, 31, 30 }, { 31, 31, 30, 31, 30, 31 } };

**String Manipulation Functions**

**Strings as Variables**

* To create a variable of type string, simply:

string fullName;

* Assignment, as always is the same:

fullName = “Aamir Hassan”;

* Or like before, we could always combine the two with an initialization:

string fullName = “Aamir Hassan”;

**Input/output with Strings**

* I/O with Strings is as before:

string fullName = “”;

cout << “Please enter your name: “;

cin >> fullName; //get the name

cout << “Your name is “ << fullName;

**String Operators: Assignment**

* Assignment (=): As with before, assign a string to a variable of type string.

string oneName = “Edwin”;

string anotherName = oneName;

* Both now hold “Edwin”

**String Operators: Concatenation**

* Concatenation (+): Puts a string on the end of another.

string firstName = “Edwin”;

string lastName = “Dreese”;

string fullName = firstName + “ ” + lastname;

//the += shorthand also works

string oneString = “2 + 2 = ”;

oneString += “5”;

**Relational String Operators**

* == and != are same as before, but the others are not exactly like usage with numbers…
* For instance, what exactly does it mean if one string is “less than” another?

**String I/O**

A common problem with reading strings from user input is that it could contain white spaces. Remember that white space (e.g. space, tab, newline) is treated as termination for cin.

* Take the following code for example:

cout << “Enter your full name: ”;

string fullname; //only the first name will be read in!!

cin >> fullname;

at(index)

This method returns the character at the specified index. Indices start from 0. • Example: string n = “Vikram”; //the character ‘V’ will be output. cout << n.at(0) << endl; Page 34 //the character ‘m’ will be output. cout << n.at(n.size()-1) << endl;

**String Processing**

* In addition to giving us a new data type to hold strings, the string library offers many useful string processing methods.
* You can find most of them of them in the book, but here are a few useful ones.

**length () and size()**

* This method returns the integer length of the string. The length() and size() are the same.
* Example:

string s1 = “Super!”;

//the integer 6 will be output.

cout << s1.length() << endl;

**at(index)**

This method returns the character at the specified index. Indices start from 0.

* Example:
* string n = “Vikram”;
* //the character ‘V’ will be output.
* cout << n.at(0) << endl;
* //the character ‘m’ will be output.
* cout << n.at(n.size()-1) << endl;

**Shorthand for at (index)**

* As an alternative, we could have also used the following equivalent shorthand:
* string n = “Vikram”;
* //the character ‘V’ will be output.
* cout << n[0] << endl;
* //the character ‘m’ will be output.
* cout << n[n.size()-1] << endl;

erase(index)

This method removes all characters from the string starting from the specified index to the end. • The length of the new string is reset to index!

* Example:
* string os = “Operating Systems”;
* os.erase(9);
* //the string “Operating” is output
* cout << os << endl;
* //length is now 9, the index
* cout << os.length() << endl;

**find(str)**

* This method returns the integer index of the first occurrence of the specified string
* Example:

string d = “data data data”;

//0 is output

cout << d.find(“data”) << endl;

**Lab Tasks:**

**Task 1:**

Write a C++ program which take the number from the user and create a dynamic array having size of a given number. This program then should print only all the even numbers. After printing, the program must free up the memory allocated to that array.

**Code:**

**#include<iostream>**

**using namespace std;**

**int main(){**

**int size;**

**cout<<"Enter the size of array : ";**

**cin>>size;**

**int arr[size];**

**cout<<endl;**

**cout<<"Enter Random numbers in the array : ";**

**for(int i=0; i<size; i++)**

**{**

**cin>>arr[i];**

**}**

**cout<<"Even numbers in the array are : ";**

**for(int i=0; i<size; i++)**

**if(arr[i]%2==0)**

**{**

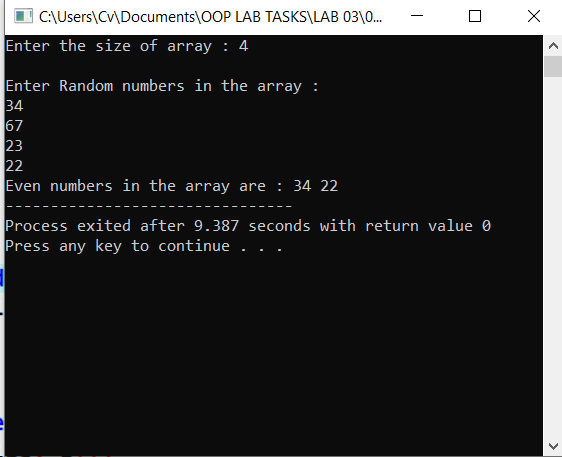
**cout<<arr[i]<<" ";**

**}**

**return 0;**

**}**

**Output:**



**Task # 02:**

Write a C++ program which extends the first task in a way that take two numbers from the user, called min and max. First the program creates the array to the size of min number and print the given numbers. After that, it dynamically resizes the array to the max number and print all the number up to the max number.

**Code:**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int min, max;**

**cout<<"Enter the minimum number : ";**

**cin>>min;**

**cout<<"Enter the maximum number : ";**

**cin>>max;**

**int min\_arr[min];**

**int max\_arr[max];**

**cout<<"Minmum numbers stored in the array are :";**

**for(int i=0; i<=min; i++)**

**{**

**min\_arr[i]=i;**

**cout<<min\_arr[i]<<" ";**

**}**

**cout<<endl;**

**cout<<"Maximum numbers stored in the array are :";**

**for(int i=min+1; i<=max; i++)**

**{**

**max\_arr[i-min]=i;**

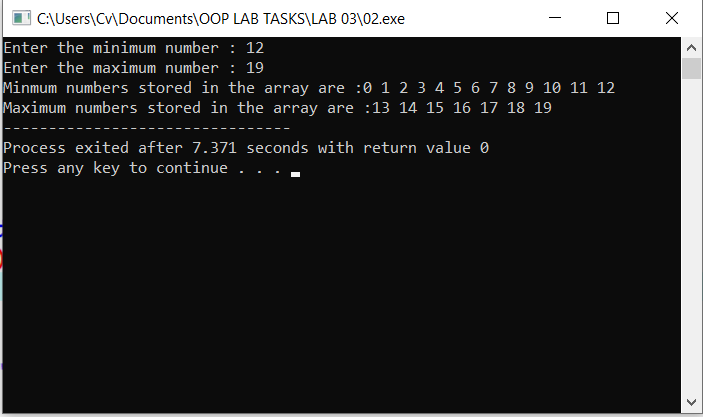
**cout<<max\_arr[i-min]<<" ";**

**}**

**return 0;**

**}**

**Output:**



**Results & Observations:**

In this Lab I’ve learned about the concept of Strings like: Reading string, displaying strings, Combining or concatenating strings, copying one string to another, comparing string & checking whether they are equal and Extraction of a portion of a string. In the first task, I’ve used an array that’s actually taking size and random numbers from the user and print out the even numbers and in the second I’ve used two arrays in which one is printing minimum numbers and other is maximum.