Fundamentals of Programing

Lab Manual # 07

**Lab Instructor:** Muhammad Affan

**Student Name: MUHAMMAD BIN AHSAN**

**CMS ID: 468098**

**DATE: 15/11/2023**

**Lab Manual # 07**

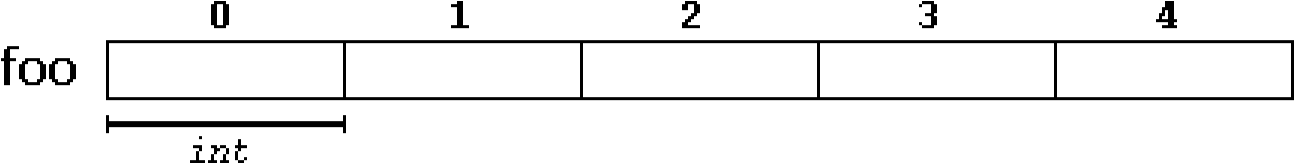
# Arrays Objective:

* To get an introduction of arrays
* Array Initialization
* Accessing array elements

# Description:

C++ provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

For example, an array containing 5 integer values of type int called foo could be represented as:



where each blank panel represents an element of the array. In this case, these are values of type int. These elements are numbered from 0 to 4, being 0 the first and 4 the last; In C++, the first element in an array is always numbered with a zero (not a one), no matter its length.

Like a regular variable, an array must be declared before it is used. A typical declaration for an array in C++ is:

**type name[no. of elements];**

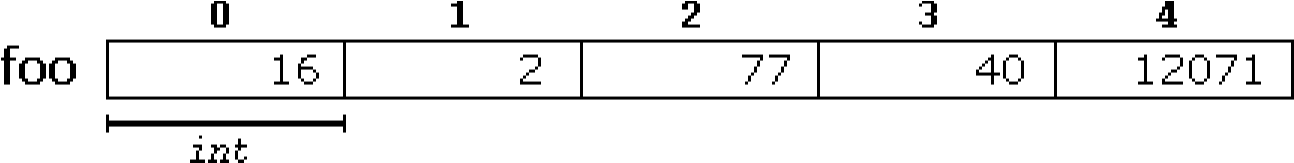
Therefore, the foo array, with five elements of type int, can be declared as:

int foo [5];

The elements in an array can be explicitly initialized to specific values when it is declared, by enclosing those initial values in braces { }. For example:

int foo [5] = { 16, 2, 77, 40, 12071 };

This statement declares an array that can be represented like this:



The number of values between braces { } cannot be greater than the number of elements in the array. For example, in the example above, foo was declared having 5 elements (as specified by the number enclosed in square brackets, [] ), and the braces { } contained exactly 5 values, one for each element. If declared with less, the remaining elements are set to their default values (which for fundamental types, means they are filled with zeroes). For example:

int bar [5] = { 10, 20, 30 };

Will create an array like this:



The initializer can even have no values, just the braces:

int bar [5] = { };

This creates an array of five int values, each initialized with a value of zero:



When an initialization of values is provided for an array, C++ allows the possibility of leaving the square brackets empty []. In this case, the compiler will assume automatically a size for the array that matches the number of values included between the braces { }:

int foo [] = { 16, 2, 77, 40, 12071 };

After this declaration, array foo would be 5 int long, since we have provided 5 initialization values.

We can access the value of any of its elements individually as if it was a normal variable, thus being able to both read and modify its value. In the case of *foo* array, one can read and write any element like this:

|  |  |
| --- | --- |
| cout<<foo[1]; | //reading index 1 |

It will display 2 as an output. Now you can modify the array elements like this:



foo[1]

=

5

;



/

/

writing

at

index



cout<<foo[1];

This time it shows 5 as an output.

One can read input in an array like this:

|  |
| --- |
| cin>>foo[1]; |

Assignment statement occurs like this:

foo[0] = bar[3];

School Of Mechanical & Manufacturing Engineering, NUST

Department of Mechanical Engineering

# Lab Task:

1. Take 10 integer inputs from user and store them in an array and print them on screen.
2. Write a program to find the sum and product of all elements of an array with 5 integer elements.
3. Print diamond pattern using a single array.

# Home Task:

1. Solve Assignment Questions related to arrays.

LAB Task 1

#include <iostream>

using namespace std;

int main() {

const int array = 10;

int numbers [array];

for (int i=0; i<10; i++) {

cout<<"Enter number "<<i+1<<endl;

cin>>numbers[i];

}

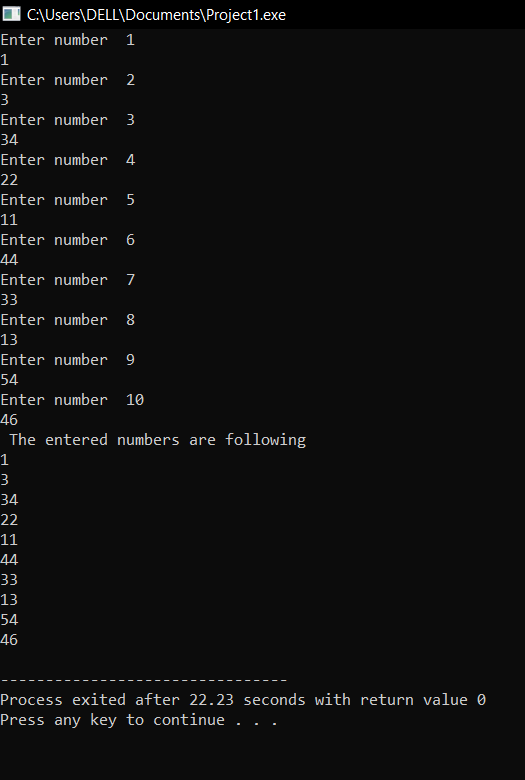
cout<<" The entered numbers are following"<<endl;

for (int i=0; i<10; i++) {

cout<<numbers[i]<<" "<<endl;

}

}



Lab Task 2

#include<iostream>

using namespace std;

int main () {

int sum = 0;

int product=1;

const int array=5;

int numbers[array];

for (int i=0; i<array; i++) {

cout<<"Enter number # "<<i+1<<endl;

cin>>numbers[i];

}

for(int i=0; i<array; i++){

sum = sum + numbers[i];

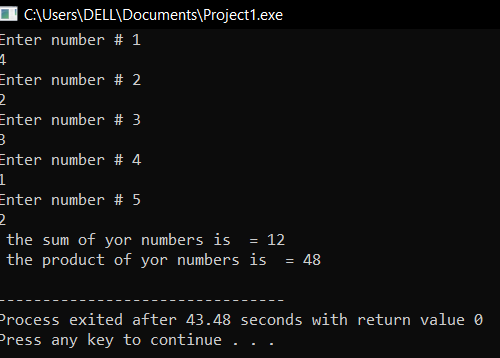
product = product\*numbers[i];

}

cout<<" the sum of yor numbers is "<<'='<<" "<<sum<<endl;

cout<<" the product of yor numbers is "<<'='<<" "<<product<<endl;

}



Lab Task 3

#include<cstring>

#include<iostream>

using namespace std;

int main()

{

int num\_rows;

cout<<"Enter the number of rows of the diamond";

cin>>num\_rows;

if(num\_rows % 2 == 0 )

{cout<<"Please enter an odd number of rows"<<endl; return 0;}

else

{

char diamond[num\_rows];

int half = num\_rows/2;

// upper half of the diamond

for(int i=0; i<=num\_rows; i++){diamond[i]=' ';}

for(int j=0; j<=half; j++)

{ diamond[half-j]='\*';

diamond[half+j]='\*';

for(int x=0; x<num\_rows; x++){cout<<diamond[x];}

cout<<endl;

}

//for lower half of diamond

for(int a=0; a<half; a++)

{diamond[num\_rows-1-a]=' ';

diamond[a]=' ';

for(int b=0; b<num\_rows; b++){cout<<diamond[b];}

cout<<endl;

}

}

return 0;

}

