School Of Mechanical & Manufacturing Engineering, NUST

Department of Mechanical Engineering

Fundamentals of Programing

Lab Manual # 08

Lab Instructor: Muhammad Affan

Student Name: _____

CMS ID: _____

DATE:

_____

Lab Manual # 08

Arrays

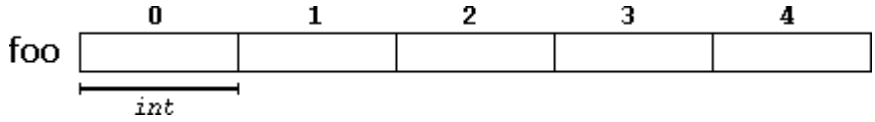
Objective:

- 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 ofthe 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 anarray 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 inC++ is:

`type name[no. of elements];`

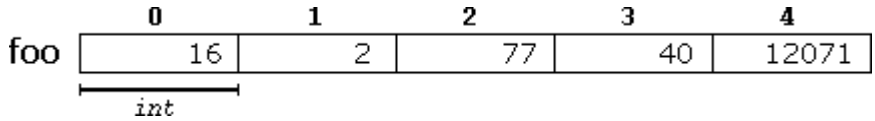
Therefore, the foo array, with five elements of type int, can be declared as:

```
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 };
```

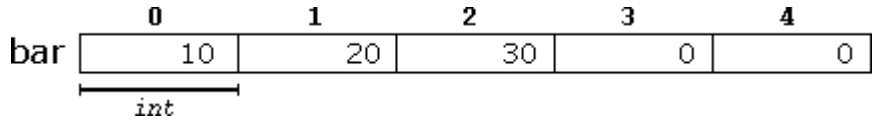
This statement declares an array that can be represented like this:



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 };
```

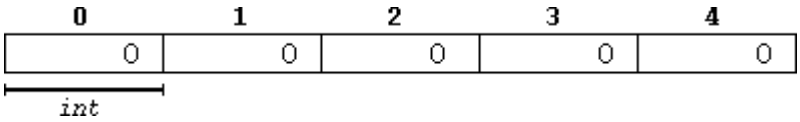
Will create an array like this:



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 squarebrackets empty []. In this case, the compiler will assume automatically a size for the array that matchesthe 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];
```

Accessing array element

```
int val = a[1];  //it will copy the value at index 1 to variable val.
```

Entering elements in array and then displaying them.

```
int main () {
    int age[4];
    for (int i=0; i < 5; i++)
     {     cout<"Enter an age : \t";
           cin>>age[i];   }
      for (int i=0; i < 5; i++)
          {     cout<"You Entered. \t"<<age[i];    }
        return 0; }
```

Lab Task:

1. Write a C++ program to calculate average of numbers of array.
2. Implement Bubble sort on an array of 5 integers.
3. Implement Selection Sort on an array of 5 integers.

Home Task:

1. Take an array and find the most repeated element in that array.
2. Let's say an array is a[8] = {13, 15, 17, 9, 99, 77, 65, 43}. Find largest and smallest element.
3. Develop a program that takes 5 array elements from user. Swap position [2] element with position [4] element. (**Hint**: Use the same method of swapping values we used for variables using a third variable temp).