Aror University of Art, Architecture, Design & Heritage Sukkur.

Department of Artificial Intelligence and Multimedia Gaming

Fundamentals of Programming (Fall-2023)

LAB No. 08

Prepared by: Abdul Haseeb Shaikh

Objective of Lab No. 08:

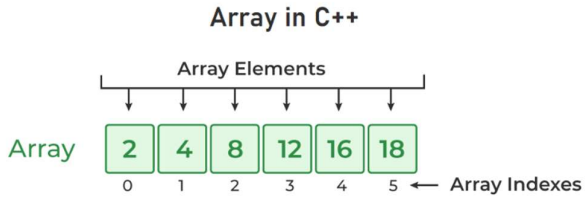
After performing lab 8, students will be able to:

- Declare, Initialize, Access and Traverse both 1D and 2D Arrays.
- Use 1D and 2D Arrays to solve real world problems

Array:

Data Structure that is used **to store multiple values** of **similar type** in a **contiguous memory** location, an array is also called as homogenous collection of data.

Array values are referred to as array elements, and each element has a position in the array called as array index which starts from 0 to n-1 where n is the size of the array.



Declaration of the array:

An array is declared using the following syntax:

```
data_type array_name[Size_of_array];
```

Example

```
int arr[5];
```

int is the data type of the array, arr is identifier name for the array, and 5 inside square brackets refers to the size of the array.

Initialization of the array:

There are multiple methods of initialization, lets discuss few of them

1. **Initialize Array with Values and size in C++:**

```
int arr[5] = {1, 2, 3, 4, 5};
```

2. **Initialize Array with Values and without Size in C++:**

```
int arr[] = {1, 2, 3, 4, 5};
```

You can also use loop to initialize the array elements with a same value or different value.

Accessing the array elements:

Elements of an array can be accessed using the index of a particular element. An example of accessing all the elements of an array using for loop is shown

below, remember that when we visit each element of an array exactly once, then that process is referred to as **Traversing the array.**

```cpp
// C++ Program to Illustrate How to Traverse an Array
#include <iostream>
using namespace std;

int main()
{

    // Initialize the array
    int table_of_two[10]
        = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };

    // Traverse the array using for loop
    for (int i = 0; i < 10; i++) {
        // Print the array elements using indexing
        cout << table_of_two[i] << " ";
    }

    return 0;
}
```

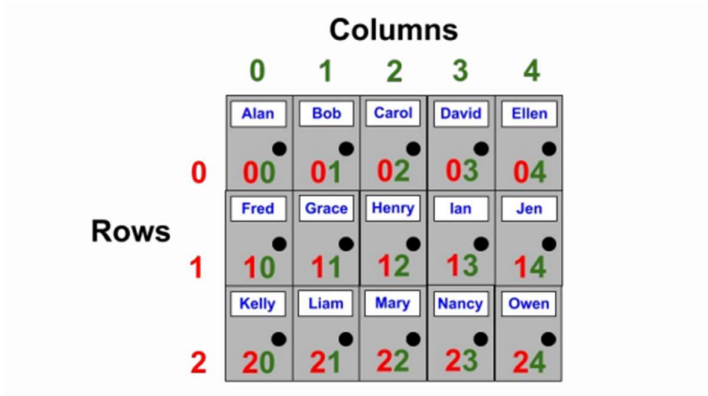
Taking Input inside an array:

```
#include<iostream>
using namespace std;

int main(){

string vegies[5];
int array_length=sizeof(vegies)/sizeof(vegies[0]);
for(int i=0; i<array_length; i++){
    cout<<"Enter Vegetable "<<i+1<<" :";
    cin>>vegies[i];
}


return 0;
```

```
Enter Vegetable 1 :Carrots
Enter Vegetable 2 :Spinach
Enter Vegetable 3 :Tomato
Enter Vegetable 4 :Potato
Enter Vegetable 5 :Corriander

--------------------------------
Process exited after 22.07 seconds with return valu
Press any key to continue . . .
```

2D Array:

- An Array inside another array
- Just like a table where there are rows and columns



Declaration of 2d Array:

```
string names[3][5];
```

Initializing and accessing 2D Array:

```
string names[ ][5] = { {"Alan", "Bob", "Carol", "David",
        "Ellen"},
                       {"Fred", "Grace", "Henry", "Ian", "Jen"},
                       {"Kelly", "Liam", "Mary", "Nancy",
        "Owen"} };

cout << names[1][2] << endl;
```

Update/Modification in 2D Array:

```
string names[3][5] = { {"Alan", "Bob", "Carol", "David",
        "Ellen"},
                       {"Fred", "Grace", "Henry", "Ian", "Jen"},
                       {"Kelly", "Liam", "Mary", "Nancy",
        "Owen"} };

cout << names[1][2] << endl;

names[1][2] = "Harry";
cout << names[1][2] << endl;
```

Traversing 2D Array:

```
int digits[3][3] = { {1, 2, 3},
                     {4, 5, 6},
                     {7, 8, 9} };

int row = sizeof(digits) / sizeof(digits[0]); //number of rows
int col = sizeof(digits[0]) / sizeof(int); // number of columns

for (int i = 0; i < row; i++) {
  for (int j = 0; j < col; j++) {
    cout << digits[i][j] << endl;
  }
}
```

Lab Exercises:

1. **Write a C++ Program to check whether an array of integers having size 10 contains a particular value or not.**

   Sample output:
   Input the element you want to search: 32
   Not Found!

2. **Write a C++ Program to test the equality of two arrays, both having size of 6 elements.**

3. **Write a C++ program to find the difference between smallest and largest element in an array of floating-point numbers having size of 15 elements.**

4. **Write a C++ program to find out the number of even and odd elements in an array of integers having size of 20 elements**

5. **Write a C++ program to add and subtract two matrices of same dimensions and print the results:**

```
Input number of rows of matrix
2
Input number of columns of matrix
2
Input elements of first matrix
1
2
3
4
Input the elements of second matrix
5
6
7
8
Sum of the matrices:-
6       8
10      12
```

6. **Write a C++ program to sum the diagonal elements of a 7*7 matrix.**

7. **Write a C++ program that grades the arithmetic quizzes as follows:**

   1. Ask the user how many questions are in the quiz.
   2. Ask the user to enter the key (that is, the correct answers). There should be one answer for each question in the quiz, and each answer should be an integer. They can be entered on a single line, e.g., 34 7 13 100 81 3 9 10 321 12 might be the key for a 10-question quiz. You will need to store the key in an array.
   3. Ask the user to enter the answers for the quiz to be graded. As for the key, these can be entered on a single line. Again there needs to be one for each question. Note that these answers do not need to be stored; each answer can simply be compared to the key as it is entered.
   4. When the user has entered all of the answers to be graded, print the number correct and the percent correct.

8. **Write a C++ program to multiply two matrices of any dimensions.**