Arrays

Ms.Kanwal Lodhi

Arrays

- Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.
- An array is a collection of elements of the same type placed in contiguous memory locations that can be individually referenced by using an index to a unique identifier.
- To declare an array, define the variable type, specify the name of the array followed by **square brackets** and specify the number of elements it should store:
- string cars[4];

Arrays (2)

Types of Arrays:

1. One dimensional array.
2. Two dimensional array.

- One dimensional array:
  - A type of array in which all elements are arranged in the form of a list is called one dimensional array. A group of elements that all share the same data type and name is what makes up a one-dimensional array.
  - It is also known as linear list.

1. Declaring : The process of specifying array name, length and data type is called array declaration.
   - type name[size];
     - where "type" is the data type, "name" is the name of the one-dimensional array, and "size" is the number of elements that the array can hold.
   - As an example:
     - int arr[5];
2. Initializing : The process of assigning values to array elements at the time of declaration.
   1. data_type array_name[array_size] = {list of values};
   - int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
   - int arr [5] = { }; \\This creates an array of five int values, each initialized with a value of zero
   - int bar [5] = { 10, 20, 30 }; \\if declared with less, the remaining elements are set to their default values (which for fundamental types, means they are filled with zeroes).

   - A syntax error occurs if the values in the braces is more then the length of array.

Arrays (3)

3. Accessing each element of an array:
   - To access an individual element of an array, use the name of the array name followed by the index of the element in square brackets.
   - Syntax: arr-name[index];
   - E.g : marks[2];

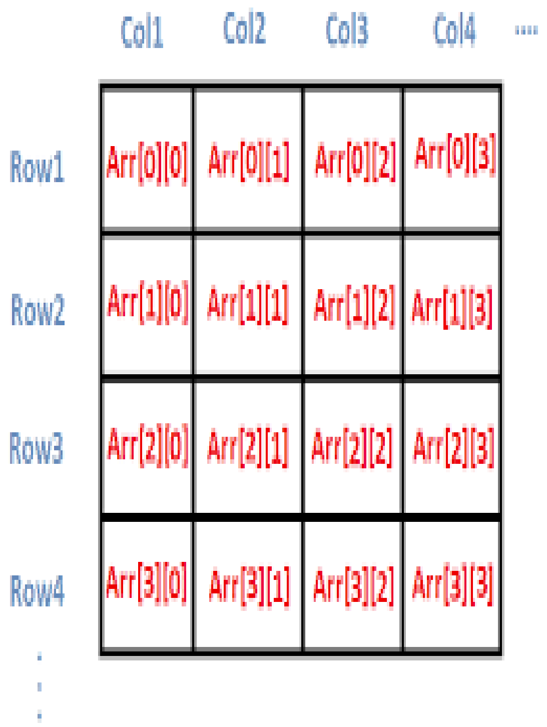
4. Accessing all array elements:
   - A easier and faster way of accessing array elements is use of loops.
   - To access each element of an array:
     1. Use a loop
     2. Let the loop variable be the index of array
     3. With each index, a different array element is accessed.
     4. Forexample:
        1.
        ```
        for(int i=0; i<=2;i++)
         cout<< marks[i] <<endl;
        ```

Arrays(4)

5. Input and output array elements:
   - cin and cout is used to input and display array elements.
   - Array index can be interger constant, integer variable or integer expression.
   - Forexample:
     - cin>>[3];   integer constant
     - cout<<[i];   integer variable
     - cout<<[i+2]   integer expression
   - The use of loop make it easier and also reduce the code.

Two dimensional array

- Two dimensional array or 2-D array can be defined as array of arrays,
- It can also represent a Matrix.
- Each element is represented as arr[row][col] , arr[][] is 2D array.
- A two-dimensional array in C++ is the simplest form of a multi-dimensional array.
- You can define one array for multiple sets of data sometimes called matrices or tables.
- Consists of row and column.
- Declaration syntax:
  - data-type arr-name[row][col];



Two Dimensional or 2D Array

- For example:
  - int arr[4][5];
    - One index for each dimension.
    - Use square bracket for each index.
    - First index represent rows and second represent columns.
    - The total no of element in each matrix can determined by multiplying rows to columns.
- Accessing 2D:
  - The array name and indexes of row and column are used to access an individual element of 2D array.
- Entering data :
  - For example; the following statement enters data in the first row of a two dimensional array:
    - Arr[0][0]=10; first row and first column.
    - Arr[0][1]=20; first row and second column.
    - Arr[1][0]=13; second row and first column.
    - Arr[1][1]=14; second row and second column.
  - Nested loops are used to enter data to two dimensional array.
  - Outer loop is used to refer to rows in array and inner loop refers to columns of 2D array.

Example of two dimensional array

```
#include<iostream>
using namespace std;
int main()
{
 int arr[2][4],i,j;
for(i=0;i<2;i++)
   for(j=0;j<4;j++)
    {
        cout<<"enter integers:\t";
    cin>>arr[i][j];
    }
for(i=0;i<2;i++)
{
  for(j=0;j<4;j++)
  cout<<arr[i][j]<<"\t";
cout<<endl;
}
return 0;
}
```

Initializing 2D arrays

- It can be initialized at the time of declaration.
- The process of initialization is done by assigning the initial value in braces separated by commas.
- The value of each row can further be assigned in nested braces.
- Example :
  - int arr[2][3]={{23,34,45},{12,56,98}};
  - int arr[2][3]={23,34,45,12,56,98};

Example :To find the max and min among integer.

```
#include<iostream>
using namespace std;
int main()
{
 int i,j,min,max;
int arr[2][3]={12,34,54,31,11,33};
max=arr[0][0];
min=arr[0][0];
for(i=0;i<2;i++)
  for(j=0;j<4;j++)
{
if(arr[i][j]>max)
 max=arr[i][j];
if(arr[i][j]<min)
min=arr[i][j];
}
cout<<"max:"<<max<<endl<<"min:"<<min;
return 0;
}
```