Assignment 4

Submission Date- 24 /10/2024

Title – Study of Array in C++.

Objective- To illustrate the Implementation of Array in C++ Programming

Problem Statement-

Implement a class CppArray which is identical to a one-dimensional C++ array (i.e. the index set is a set of consecutive integers starting at 0) except for the following:

1. It performs range checking.

2. It allows one to be assigned to another array through the use of assignment operator.

3. It supports a function that returns the size of the array.

Software & Hardware requirements- any Text editor and Terminal in Linux/ Turbo C++ Compiler installed on PC.

Theory-

Math Library Functions

C++ supports a large number of useful mathematical functions. These functions are available in standard C++ to support various mathematical calculations. In order to use these functions you need to include a header file- **<math.h>**

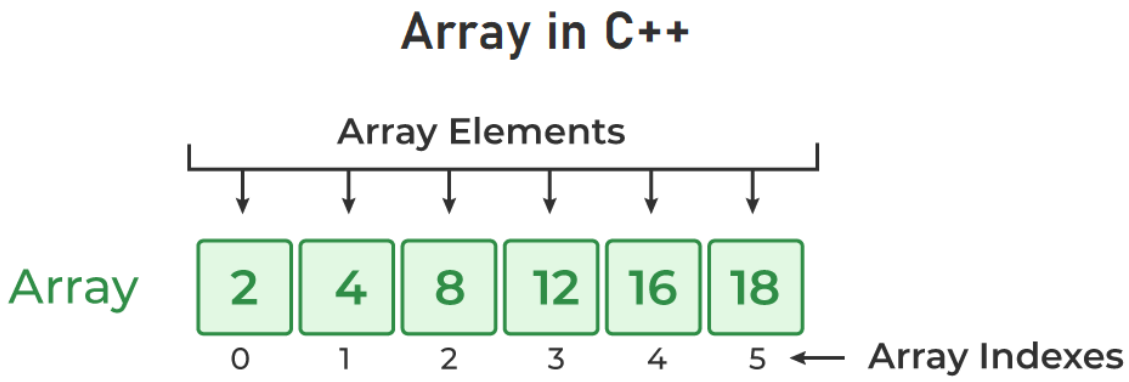
#include<math.h> or <cmath>

Math Library Functions in C++

1. max(*x*,*y*) function can be used to find the highest value of *x* and *y*
2. min(*x*,*y*) function can be used to find the lowest value of *x* and *y*
3. sqrt (square root), round (rounds a number) and log (natural logarithm), can be found in the <cmath> header file.

Array –

In C++, an array is a data structure that is used to store multiple values of similar data types in a contiguous memory location.



Properties of Arrays in C++

- An Array is a collection of data of the same data type, stored at a contiguous memory location.
- Indexing of an array starts from 0. It means the first element is stored at the 0th index, the second at 1st, and so on.
- Elements of an array can be accessed using their indices.
- Once an array is declared its size remains constant throughout the program.
- An array can have multiple dimensions.
- The size of the array in bytes can be determined by the sizeof operator using which we can also find the number of elements in the array.
- We can find the size of the type of elements stored in an array by subtracting adjacent addresses.

Array Declaration in C++

In C++, we can declare an array by simply specifying the data type first and then the name of an array with its size.

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

Example

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

Initialization of Array in C++

In C++, we can initialize an array in many ways. We can initialize an array at the time of declaration or after declaration.

1. Initialize Array with Values in C++

We have initialized the array with values. The values enclosed in curly braces '{}' are assigned to the array. Here, 1 is stored in arr[0], 2 in arr[1], and so on. Here the size of the array is 5.

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

Accessing an Element of an Array in C++

Elements of an array can be accessed by specifying the name of the array, then the index of the element enclosed in the array subscript operator [ ]. For example, arr[i].

Size of an Array in C++

We can calculate the size of an array using sizeof ( ) operator . First, we find the size occupied by the whole array in the memory and then divide it by the size of the type of element stored in the array. This will give us the number of elements stored in the array.

```
data_type size = sizeof(Array_name) / sizeof(Array_name[index]);
```

Code

```
#include<iostream>
#include<math.h>
using namespace std;

class array
{
    int a[10],b[10],i,j,temp,n;

   public:
          void accept();
          void display();
          void range();
          void sort();
          void exchange();
          int size();
};

void array::accept()
{
   cout<<"How many Elements you want to insert into the array=\n";
   cin>>n;
   cout<<"Enter the array Elements=\n";
   for(i=0;i<n; i++)
      {
          cout<<"a["<<i<<"]=";
          cin>>a[i];
       }
}

void array::display()
{
```

```
    cout<<"For Enter Array Elements are=\n";
     for(i=0;i<n; i++)
      {
         cout<<"a["<<i<<"]="<<a[i]<<"\n";
      }
}

void array::range()
{
    cout<<"The range of array is from"<<a[0]<<"to"<<a[n-1]<<endl;
}

void array::sort()
{
    for(i=0;i<n; i++)
    {
        for(j=0;j<n-1;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }

   cout<<"\n The Elements of Array After sorting";
    for(i=0;i<n; i++)
    {
       cout<<"a["<<i<<"]="<<a[i]<<"\n";
    }
}
```

```
void array::exchange()
{
    for(i=0;i<n; i++)
     {
         b[i]=a[i];
     }

   cout<<"\n The array Elements Array exchanging=";
   for(i=0;i<n; i++)
     {
       cout<<"b["<<i<<"]="<<b[i]<<"\n";
     }
}

int array::size()
{
    return n;
}

int main()
{
     int x;
     array obj;
     obj.accept();
     obj.display();
     obj.range();
     obj.sort();
     obj.exchange();
     x=obj.size();

  cout<<"\n The size of Array="<<x; ***
  return 0;
```

}

Output

```
[student@localhost P]$ g++ array.cpp
[student@localhost P]$ ./a.out
How many Elements you want to insert into the array=
5
Enter the array Elements=
a[0]=12
a[1]=11
a[2]=32
a[3]=23
a[4]=45
For Enter Array Elements are=
a[0]=12
a[1]=11
a[2]=32
a[3]=23
a[4]=45
The range of array is from12to45

The Elements of Array After sorting a[0]=11
a[1]=12
a[2]=23
a[3]=32
a[4]=45

The array Elements Array exchanging= b[0]=11
b[1]=12
b[2]=23
b[3]=32
```

b[4]=45

The size of Array=5 [student@localhost P]$

Conclusion-

Understanding concept of Array and their usage of Array in C++.