

## 6-1 Arrays:

An array is a consecutive group of homogeneous memory locations. Each element (location) can be referred to using the array name along with an integer that denotes the relative position of that element within the array.

### 6-1 Array of One Dimension:

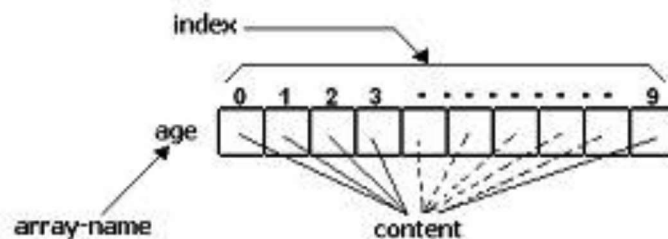
#### 1 Declaration of 1D-Arrays:

##### **General Form of 1D-Array:**

```
data-type Array-name [ size ];
```

##### Examples:

```
int    age [10];  
int    num [30];  
float  degree[5];  
char   a [15];
```



#### 2 Initializing Array Elements:

- The first element of array age:

```
age [0] = 18;
```

- The last element of array age:

```
age [9] = 19;
```

- All elements of array age:

```
age [9] = { 18, 17, 18, 18, 19, 20, 17, 18, 19 };
```

- int x [ ] = { 12, 3, 5, 0, 11, 7, 30, 100, 22 };

- int y [10] = { 8, 10, 13, 15, 0, 1, 17, 22 };

### 3 Accessing Array Elements:

- Accessing the first element of array num to variable x:

```
x = num [0];
```

- Accessing the last element of array num to variable y:

```
y = num [9];
```

- cout << num [0] + num [9];

- num [0] = num [1] + num[2];

- num [7] = num [7] + 3;                      num [7] += 3;

### 4 Read / Write / Process Array Elements:

- cout << num [4];

```
- for (int i=0; i<10; i++)  
    cout << num[ i ];
```

```
- if ( num [5] > 5 )  
    cout << "greater";
```

```
- for (int i=9; i>=0; i++)  
    cout << num[ i ];
```

```
- for (int i=0; i<10; i++)  
    cin >> num[ i ];
```

```
- sum=0;  
for (int i=0; i<10; i++)  
    sum = sum + num[ i ];
```

#### **Example 6.1**

\_\_ Write C++ program to display 2<sup>nd</sup> and 5<sup>th</sup> elements of array distance:

```
#include<iostream.h>
```

```
void main( )
```

```
{
```

```
    double distance[ ] = { 23.14, 70.52, 104.08, 468.78, 6.28};
```

```
    cout << "2nd element is: " << distance[1] << endl;
```

```
    cout << "5th element is: " << distance[4];
```

```
}
```

#### **Example 6.2**

\_\_ Write C++ program to read 5 numbers and print it in reverse order:

```
#include<iostream.h>
```

```
void main( )
```

```
{
```

```
    int a [5];
```

```
    cout << "Enter 5 numbers \n";
```

```

for ( int i=0; i <5; i++ )
{
    cout << i << ": ";
    cin >> a [ i ];
    cout << "\n";
}
cout << "The reverse order is: \n";
for ( i =4; i >=0; i-- )
    cout << i << ": " << a [ i ] << endl;
}

```

### Example 6.3

— Write C++ program, to find the summation of array elements:

```

#include<iostream.h>
void main ( )
{
    int const L = 10;
    int a [L];
    int sum = 0;
    cout << "enter 10 numbers \n";
    for ( int i =0; i <L; i++ )
    {
        cout << "enter value " << i << ": ";
        cin >> a [ i ];
        sum += a [ i ];
    }
    cout << "sum is: " << sum << endl;
}

```

### Example 6.4

— Write C++ program, to find the minimum value in array of 8 numbers:

```

#include<iostream.h>
void main ( )
{
    int n = 8;
    int a [ ] = { 18, 25, 36, 44, 12, 60, 75, 89 };
    int min = a [ 0 ];
    for ( int i = 0; i < n; i++ )
        if ( a [ i ] < min )    min = a [ i ];
    cout << "The minimum number in array is: " << min;
}

```

### Example 6.5

— Write C++ program, using function, to find (search) X value in array, and return the index of it's location:

```
#include<iostream.h>

int search( int a[ ], int y)
{
    int i= 0;
    while ( a [ i ] != y )
        i++;
    return ( i );
}

void main ( )
{
    int X, f;
    int a [ 10 ] = { 18, 25, 36, 44, 12, 60, 75, 89, 10, 50 };
    cout << "enter value to find it: ";
    cin >> X;
    f= search (a, X);
    cout << "the value " << X << " is found in location "<< f;
}

```

### Apply it:

```
#include<iostream.h>
void main ( )
{
    int X, i= 0, found=0;
    int a [ 10 ] = { 18, 25, 36, 44, 12, 60, 75, 89, 10, 50 };
    cout << "enter value to find it: ";
    cin >> X;
    while (( a [ i ] != X ) && ( i < 10 ))
        i++;
    if (i < 10) found=1;
    else found=0;

    if (found == 1 )
        cout << "the value " << X << " is found in location "<< i;
    else cout << "the value " << X << " is not found";
}

```

Exercise: Rewrite the above program using function.

### Example 6.6

\_\_ Write C++ program, to split the odd numbers and even numbers of one array into two arrays:

```
a = [ 1, 2, 3, 4, 5, 6, 7, 8, ... , 20 ]  
aodd = [ 1, 3, 5, 7, ... , 19 ]  
aeven = [ 2, 4, 6, 8, ... , 20 ]
```

```
#include<iostream.h>
```

```
void main ( )
```

```
{
```

```
    int a [ 20 ]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 };
```

```
    int aodd[20], aeven [20];
```

```
    int i ,o=0, e=0;
```

```
    for ( i=0 ; i<20; i++ )
```

```
        if (a[i] % 2 !=0)
```

```
        {
```

```
            aodd[o]=a[i];
```

```
            o=o+1;
```

```
        }
```

```
        else
```

```
        {
```

```
            aeven[e]=a[i];
```

```
            e=e+1;
```

```
        }
```

```
    for ( i=0 ; i<o; i++ )
```

```
        cout<<aodd[i]<<" ";
```

```
    cout<<endl;
```

```
    for ( i=0 ; i<e; i++ )
```

```
        cout<<aeven[i]<<" ";
```

```
}
```