

C++ Arrays — Beginner Friendly Notes

1. Definition

An array is a collection of elements of the same data type stored in continuous memory locations and accessed using

Example:

```
int marks[5] = {80, 75, 90, 60, 85};
```

2. Why We Use Arrays

- Store many values using one variable name
- Easy to access data using index
- Saves memory and time
- Useful in loops and data processing

3. Basic Syntax of Array

```
data_type array_name[size];
```

Example:

```
int numbers[5];
```

```
float price[10];
```

```
char grade[3];
```

4. Declaring an Array

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

5. Declaring and Initializing Array

Method 1:

```
int arr[5] = {10, 20, 30, 40, 50};
```

Method 2:

```
int arr[] = {10, 20, 30, 40};
```

Method 3:

```
int arr[5] = {10, 20};
```

6. Accessing Array Elements

Index starts from 0

```
int arr[5] = {10,20,30,40,50};
```

```
cout << arr[0];
```

```
cout << arr[2];
```

7. Input Values in Array

```
#include<iostream>
```

```
using namespace std;
```

```
int main(){
```

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

```
    for(int i=0;i<5;i++){
```

```
        cin>>arr[i];
```

```
    }
```

```
    for(int i=0;i<5;i++){
```

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

```
    }
```

```
    return 0;
```

```
}
```

8. Changing Array Value

```
int arr[3] = {10,20,30};
```

```
arr[1] = 50;
```

9. Types of Arrays

One Dimensional:

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

Two Dimensional:

```
int arr[3][3];
```

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

10. Important Rules

- Array size must be fixed

- Index starts from 0

- Same data type only
- Wrong index causes error

11. Real Life Example

```
int marks[5] = {85,90,78,92,88};
```

```
float temp[7] = {36.5,37.0,36.8,37.2,36.9,37.1,36.7};
```

12. Summary

- Store multiple values
- Same data type
- Access using index
- Very important in C++