**C++ Arrays**

## **C++ Single Dimensional Array**

Let's see a simple example of C++ array, where we are going to create, initialize and traverse array.

1. #include <iostream>
2. **using** **namespace** std;
3. **int** main()
4. {
5. **int** arr[5]={10, 0, 20, 0, 30};  //creating and initializing array
6. //traversing array
7. **for** (**int** i = 0; i < 5; i++)
8. {
9. cout<<arr[i]<<"\n";
10. }
11. }

Output:

10

0

20

0

30

## **C++ Array Example: Traversal using foreach loop**

We can also traverse the array elements using foreach loop. It returns array element one by one.

1. #include <iostream>
2. **using** **namespace** std;
3. **int** main()
4. {
5. **int** arr[5]={10, 0, 20, 0, 30}; //creating and initializing array
6. //traversing array
7. **for** (**int** i: arr)
8. {
9. cout<<i<<"\n";
10. }
11. }

Output:

10

20

30

40

50

## **Passing array as a parameter of a method’s function**

#include <iostream>

using namespace std;

void sex(int nipple[500], int capacity){

for (int i = 0; i < capacity; i++) {

cout << nipple[i] << "\n";

}

}

int main(){

cout << "no of elements in the array" << endl;

int x;

cin >> x;

int array[x];

for (int i = 0; i < x; i++){

cin >> array[i];

}

sex(array, x);

return (0);

}

## **Passing array as a parameter of a method and returning value as well**

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;  int sex(int nipple[500], int capacity){  for (int i = 0; i < capacity; i++) {  cout << nipple[i] << "\n";  }  //finding the max element  int min = nipple[0];  for (int i = 0; i < capacity; i++) {  if (min > nipple[i]){  min = nipple[i];  }  }  return min;  }  int main(){  cout << "no of elements in the array" << endl;  int x;  cin >> x;  int array[x];  for (int i = 0; i < x; i++){  cin >> array[i];  }  int take = sex(array, x);    cout << "lowest fuckin value will be " << take;    return (0);  } | no of elements in the array  10  50  6  7  -2  8  6  15  21  3  0  50  6  7  -2  8  6  15  21  3  0  lowest fuckin value will be -2 |

**Multidimensional array**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #include <iostream>  using namespace std;  int main()  {  int test[3][3]; //declaration of 2D array  test[0][0]=5; //initialization  test[0][1]=10;  test[0][2]=25;  test[1][1]=15;  // test[1][2]=20;  test[2][0]=30;  test[2][2]=10;  //traversal  for(int i = 0; i < 3; ++i)  {  for(int j = 0; j < 3; ++j)  {  cout<< test[i][j]<<" ";  }  cout<<"\n"; //new line at each row  }  return 0;  } | | | 5 10 25  0 15 0  30 0 10 | |
| #include <iostream>  using namespace std;  int main()  {  //declaration and initialization  int test[3][3] =  {  {2, 5, 5},  {4, 0, 3},  {9, 1, 8}  };  //traversal  for(int i = 0; i < 3; ++i)  {  for(int j = 0; j < 3; ++j)  {  cout<< test[i][j]<<" ";  }  cout<<"\n"; //new line at each row  }  return 0;  } | | | 2 5 5  4 0 3  9 1 8 | |
| **If we don’t supply any one parameter** | | |  | |
| int test[3][3] =  {  {2, 5},  {4, 0, 3},  {9, 1, 8}  }; | Although we tried to clear the 2nd value from 2nd row, but THE CHANGE HAD HAPPENED AT THE LAST ELEMENT, THAT TURBNED TO 0 | int test[3][3] =  {  {2, 5,},  {4, 3},  {9, 1, 8}  }; | 2 5 0  4 0 3  9 1 8 | 2 5 5  4 3 0  9 1 8 |