

Create and Delete

Below, we examine the different methods available for constructing/destroying containers with particular parameters.

You can construct each container by a multitude of constructors. To delete all elements of a container `cont`, you can use `cont.clear()`. It makes no difference if you create a container, if you delete them or if you add or remove elements. Each time the container takes care of the memory management.

The table shows you the constructors and destructors of a container. A `std::vector` stands for the rest of them.

Type	Example
Default	<code>std::vector<int> vec1</code>
Range	<code>std::vector<int> vec2(vec1.begin(), vec1.end())</code>
Copy	<code>std::vector<int> vec3(vec2)</code>
Copy	<code>std::vector<int> vec3= vec2</code>
Move	<code>std::vector<int> vec4(std::move(vec3))</code>
Move	<code>std::vector<int> vec4= std::move(vec3)</code>
Sequence (Initializer list)	<code>std::vector<int> vec5 {1, 2, 3, 4, 5}</code>

Sequence (Initializer list)	<code>std::vector<int> vec5= {1, 2, 3, 4, 5}</code>
Destructor	<code>vec5.~vector()</code>
Delete elements	<code>vec5.clear()</code>

Creation and deletion of a container

Because `std::array` is generated at compile time, there are a few things that are special. `std::array` has no move constructor and can neither be created with a range nor with an initialiser list. However, you can initialize a `std::array` with an aggregate initialization. Also, `std::array` has no method for removing its elements.

Now I can use the different constructors on the different containers.

```
// containerConstructor.cpp
#include <iostream>
#include <map>
#include <unordered_map>
#include <vector>
using namespace std;

int main(){
    vector<int> vec= {1, 2, 3, 4, 5, 6, 7, 8, 9};
    map<string, int> m= {"bart", 12345}, {"jenne", 34929}, {"huber", 840284} };
    unordered_map<string, int> um{m.begin(), m.end()};

    for (auto v: vec) cout << v << " "; // 1 2 3 4 5 6 7 8 9
    cout << "\n";
    for (auto p: m) cout << p.first << "," << p.second << " "; //bart,12345 huber,840284 jenne,
    cout << "\n";
    for (auto p: um) cout << p.first << "," << p.second << " "; //bart,12345 jenne,34929 huber,
    cout << "\n";

    vector<int> vec2= vec;
    cout << vec.size() << endl; // 9
    cout << vec2.size() << endl; // 9

    vector<int> vec3= move(vec);
    cout << vec.size() << endl; // 0
    cout << vec3.size() << endl; // 9

    vec3.clear();
    cout << vec3.size() << endl; // 0
    return 0;
```

}



Various constructors