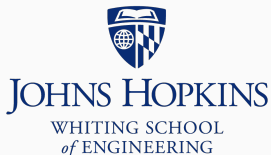


Ben Langmead

ben.langmead@gmail.com

www.langmead-lab.org



Source markdown available at github.com/BenLangmead/c-cpp-notes

Collection of *keys*, each with an associated *value*

- Like Java's `java.util.HashMap` or `TreeMap`
- Like Python's `dict` (dictionary) type

Value can be any type you wish (even another container)

Key can be any type where `<` can be used to compare values

- All numeric types, `char`, `std::string`, etc

C++: map

Declare a map:

```
map<int, string> jhed_to_name;
```

Add a key + value to a map:

```
jhed_to_name[92394] = "Alex Hamilton";
```

Print a key and associated value:

```
const int k = 92394;  
cout << "Key=" << k << ", Value=" << jhed_to_name[k] << endl;
```

A map can only associate 1 value with a key

```
const int k = 92394;
jhed_to_name[k] = "Alex Hamilton";
jhed_to_name[k] = "George Washington"; // Alex is replaced
cout << k << ": " << jhed_to_name[k] << endl;
```

```
$ g++ -c jhed_map_0.cpp -std=c++11 -pedantic -Wall -Wextra
```

```
$ g++ -o jhed_map_0 jhed_map_0.o
```

```
$ ./jhed_map_0
```

```
92394: George Washington
```

Get number of keys:

```
jhed_to_name.size();
```

Check if map contains (has a value for) a given key:

```
if(jhed_to_name.find(92394) != jhed_to_name.end()) {  
    cout << "Found it" << endl;  
} else {  
    cout << "Didn't find it" << endl;  
}
```

C++: vector

To get all the elements of the map, use an *iterator*:

```
for(map<int, string>::iterator it = jhed_to_name.begin();
    it != jhed_to_name.end();
    ++it)
{
    cout << " " << it->first << ": " << it->second << endl;
}
```

Iterator type: `map<int, string>::iterator`

Loop is similar to the loop for vector

Keys iterated over *in ascending order* (increasing id in this case)

Looking at the body:

```
cout << " " << it->first << ": " << it->second << endl;
```

Dereferenced map iterator type is `std::pair<key_type, value_type>`

- `it->first` is the key (int here)
- `it->second` is the value (string here)

More on pair later

C++: map

```
#include <iostream>
#include <map>
#include <string>
using std::cout;    using std::endl;
using std::string;  using std::map;

int main() {
    map<int, string> jhed_to_name;
    jhed_to_name[92394] = "Alex Hamilton";
    jhed_to_name[13522] = "Ben Franklin";
    jhed_to_name[42345] = "George Washington";
    cout << "size of jhed_to_name " << jhed_to_name.size() << endl;
    cout << "jhed_to_name[92394] " << jhed_to_name[92394] << endl;
    for(map<int, string>::iterator it = jhed_to_name.begin();
        it != jhed_to_name.end();
        ++it)
    {
        cout << " " << it->first << ": " << it->second << endl;
    }
    return 0;
}
```


C++: map

```
$ g++ -c jhed_map.cpp -std=c++11 -pedantic -Wall -Wextra
$ g++ -o jhed_map jhed_map.o
$ ./jhed_map
size of jhed_to_name 3
jhed_to_name[92394] Alex Hamilton
  13522: Ben Franklin
  42345: George Washington
  92394: Alex Hamilton
```

Note again: the keys are printed *in ascending order*

Can use `reverse_iterator`, `.rbegin()` and `.rend()` to get keys in *descending order*