

How would I loop through a `std::map` in C++? My map is defined as:

```
std::map<std::string, std::map<std::string, std::string>>
```

For example, this holds data like this:

```
m["name1"]["value1"] = "data1";  
m["name1"]["value2"] = "data2";  
m["name2"]["value1"] = "data1";  
m["name2"]["value2"] = "data2";  
m["name3"]["value1"] = "data1";  
m["name3"]["value2"] = "data2";
```

you can use a [ranged based for loop](#) and simply do:

```
std::map<std::string, std::map<std::string, std::string>> mymap;  
  
for(auto const &ent1 : mymap) {  
    // ent1.first is the first key  
    for(auto const &ent2 : ent1.second) {  
        // ent2.first is the second key  
        // ent2.second is the data  
    }  
}
```

this should be much cleaner than the earlier versions, and avoids unnecessary copies.

Some favour replacing the comments with explicit definitions of reference variables (which get optimised away if unused):

```
for(auto const &ent1 : mymap) {  
    auto const &outer_key = ent1.first;  
    auto const &inner_map = ent1.second;  
    for(auto const &ent2 : inner_map) {  
        auto const &inner_key = ent2.first;  
        auto const &inner_value = ent2.second;  
    }  
}
```

From:

<https://stackoverflow.com/questions/4844886/how-to-loop-through-a-c-map-of-maps>