

POGIL Activity 28: C++ Maps

Introduction to Maps

- Maps store key-value pairs and automatically sort them by the key.
- Operations include insertion, retrieval, and iteration.

Learning Objectives

- Understand the structure and usage of `std::map` in C++.
- Learn to perform key-value operations in maps.
- Compare the order of elements in `std::map` with Python dictionaries.

Critical Thinking Questions

1. How do C++ maps compare to Python dictionaries in terms of structure and usage?
2. What are the benefits of using maps in C++ for key-value storage?
3. How does map iteration work in C++?

Alternate C++ IDE: <https://www.programiz.com/cpp-programming/online-compiler/>

Role	Name	Email
Facilitator		
Spokesperson		
Quality Control		
Process Analyst		

Setup

1. The **spokesperson** should make a copy of this document in your own folder and then share the new document with professor Aydin (aydinn@kenyon.edu), the graders oppongkrampah1@kenyon.edu, coons1@kenyon.edu with edit access. Notify everyone.
2. The **facilitator** should create a new project in an online IDE of your choice. Paste link to the project in the box below after saving it. Make sure to save your code after editing it.
3. Have the spokesperson enter in your answers for each item below.
4. Use a different color to enter your answers

Links to your projects	
------------------------	--

Hands-On Activities

Basic Map Usage: Run the following code to explore basic map operations:

```
#include <iostream>
#include <map>
```

```
using namespace std;
```

```
int main() {  
    map<string, int> grades;  
    grades["Emily"] = 95;  
    grades["Sam"] = 88;  
  
    cout << "Sam's grade: " << grades["Sam"] << endl;  
    cout << "Emily's grade: " << grades["Emily"] << endl;  
  
    return 0;  
}
```

1. Questions

- What Python data structure is this similar to?
- What are the data types of the keys and values in the example above?

Iterating Through a Map Add the following to the program to iterate through the map:

```
cout << "All grades: " << endl;  
for (auto gradePair : grades) {  
    cout << gradePair.first << ": " << gradePair.second << endl;  
}  
/* Here is a more traditional C++ way of doing the same thing, using an iterator. Again, to learn more  
about iterators, take Comp 218.  
  
map<string, int>::iterator it;  
for(it=grades.begin(); it!=grades.end(); it++){  
    cout<<(*it).first << ": " << (*it).second<< endl;  
}  
*/
```

2. Questions

- What does `gradePair.first` and `gradePair.second` return?
- How does the iteration order compare to Python dictionaries?

Map of Vectors: Explore more advanced operations by running this:

```
#include <iostream>  
#include <map>
```

```

#include <vector>
using namespace std;

int main() {
    map<string, vector<int>> studentGrades;

    studentGrades["Alice"] = {90, 85, 88};
    studentGrades["Bob"] = {78, 82, 80};

    for (auto& pair : studentGrades) {
        cout << pair.first << "'s grades: ";
        for (int grade : pair.second) cout << grade << " ";
        cout << endl;
    }

    return 0;
}

```

3. Questions

- Explain the syntax `map<string, vector<int>>`.
- What are potential use cases for a map of vectors?

4. Interactive Program

Modify the program to allow user input:

- Prompt the user to enter a student's name and their grades. Let the user enter as many names as they wish until they enter -1. Store the data in the map and display all stored grades.

5. Questions

- How does this modification enhance functionality?
 - Compare this use case with Python dictionaries and lists.
-

Extension: Maps, Vectors, and Classes

1. Consider the following code that uses a class to store a student name and a grade list:
<https://onlinegdb.com/WM6Ez6rMk>
2. Modify the program to use `map<string, Student> students` to store a Map which maps from the student's name to the associated student objects. Use the following code:
<https://codeboard.io/projects/471591>

3. Modify the code to, instead of listing the students after they are entered, add a loop that asks for a student name, and looks up and displays that student, then allows an additional grade to be added for that student, e.g.: <https://codeboard.io/projects/471612>

Insert here links to your completed projects.

Reflection

Discuss how `std::map` differs from Python dictionaries in terms of order and use cases.