Size

Let's find out how we can check the size and capacity of C++ containers.

For a container `cont`, use `cont.empty()` to see if the container is empty. `cont.size()` returns the current number of elements, and `cont.max_size()` returns the maximum number of elements `cont` can have. The maximum number of elements is implementation defined.

```
// containerSize.cpp
#include <iostream>
#include <map>
#include <set>
#include <vector>

using namespace std;

int main(){
  vector<int> intVec{1, 2, 3, 4, 5, 6, 7, 8, 9};
  map<string, int> str2Int = {{"bart", 12345},
                              {"jenne", 34929}, {"huber", 840284}};
  set<double> douSet{3.14, 2.5};

  cout << intVec.empty() << endl;  // false
  cout << str2Int.empty() << endl; // false
  cout << douSet.empty() << endl;  // false

  cout << intVec.size() << endl;  // 9
  cout << str2Int.size() << endl; // 3
  cout << douSet.size() << endl;  // 2

  cout << intVec.max_size() << endl;  // 4611686018427387903
  cout << str2Int.max_size() << endl; // 256204778801521550
  cout << douSet.max_size() << endl;  // 461168601842738790
  return 0;
}
```

Size of a container

🔑 Use cont.empty() instead of cont.size()

For a container `cont`, use the method `cont.empty()` instead of

`(cont.size() == 0)` to determine if the container is empty. First,

`cont.empty()` is, in general, faster than `(const.size() == 0)`; second, the container `std::forward_list` has no method `size()`.

In the next lesson, we'll discuss how to access the elements of a container.