Insertion and Deletion

In associative containers, values are inserted and deleted based on the keys they have.

The insertion (`insert` and `emplace`) and deletion (`erase`) of elements in associative containers is similar to the rules of an std::vector. For an associative container that can have a key only once, the insertion fails if the key is already in the container. Additionally, ordered associative containers support a special function `ordAssCont.erase(key)`, which removes all pairs with the `key` and returns their number. See the usage of the function below.

```
// associativeContainerModify.cpp
#include <iostream>
#include <set>
#include <array>

int main(){
  std::multiset<int> mySet{3, 1, 5, 3, 4, 5, 1, 4, 4, 3, 2, 2, 7, 6, 4, 3, 6};

  for (auto s: mySet) std::cout << s << " "; // 1 1 2 2 3 3 3 3 4 4 4 4 5 5 6 6 7
  std::cout << "\n";

  mySet.insert(8);
  std::array<int, 5> myArr{10, 11, 12, 13, 14};
  mySet.insert(myArr.begin(), myArr.begin()+3);
  mySet.insert({22, 21, 20});
  for (auto s: mySet) std::cout << s << " ";
  // 1 1 2 2 3 3 3 3 4 4 4 4 5 5 6 6 7 10 11 12 20 21 22
  std::cout << "\n";

  std::cout << mySet.erase(4);  // 4
  mySet.erase(mySet.lower_bound(5), mySet.upper_bound(15));
  for (auto s: mySet) std::cout << s << " ";
    // 1 1 2 2 3 3 3 3 20 21 22
  std::cout << "\n";

  return 0;
}
```

In the next two chapters, we will learn about the two kinds of associative

containers:

- ordered associative containers
- unordered associative containers