Sets 和 Maps 的介绍及示例

1. Sets 的特点和用法

特点

- `set` 是一种有序容器，其中的元素按键值排序，并且不允许重复。
- 常见操作包括插入、删除和查找，时间复杂度为 $O(\log n)$。
- 提供迭代器 (`iterator`, `const_iterator`) 用于遍历容器。

示例代码：基本使用

Listing 1: Set 示例

```
#include <iostream>
#include <set>
using namespace std;

int main() {
    set<int> mySet;

    // 48橡橡时
    mySet.insert(10);
    mySet.insert(20);
    mySet.insert(10); // 48橡橡时氈橡时

    // 48橡橡时
    for (int val : mySet) {
        cout << val << "-"; // 48橡                    : 10 20
    }
    cout << endl;
```

```
    // 48鬟橡时
    if (mySet.find(20) != mySet.end()) {
        cout << "20 48橡  簧橡                    " << endl;
    }

    //48簧橡时
    mySet.erase(10);
    cout << "48橡橡橡                  : " << mySet.size() << endl;

    return 0;
}
```

自定义排序规则

通过比较函数自定义排序，例如降序：

Listing 2: Set 自定义排序

```
struct Compare {
    bool operator()(int a, int b) const {
        return a > b; // 48橡
    }
};

set<int, Compare> mySet = {10, 20, 5};
for (int val : mySet) {
    cout << val << " "; // 48橡                  : 20 10 5
}
```

2. Maps 的特点和用法

特点

- `map` 是一种有序的键值对容器，键（`key`）唯一，值（`value`）可以重复。
- 支持键到值的高效查找，操作复杂度为 $O(\log n)$。
- 提供类似数组的索引操作符 `[]`。

示例代码：基本使用

Listing 3: Map 示例

```
#include <iostream>
#include <map>
using namespace std;

int main() {
    map<string, int> myMap;

    // 48檫簣檫檫
    myMap["Alice"] = 30;
    myMap["Bob"] = 25;

    // 48檫簣檫檫
    for (auto &entry : myMap) {
        cout << entry.first << ":" << entry.second << endl;
    }

    // 48髮簣
    if (myMap.find("Alice") != myMap.end()) {
        cout << "Alice 48檫 髮檫                    " <<endl;
    }

    //48簣簣
    myMap.erase("Bob");
    cout << "Alice 48檫簣          : " <<myMap["Alice"] <<endl;

    return 0;
}
```

自定义比较规则

Listing 4: Map 自定义排序

```
struct Compare {
    bool operator()(const string &a, const string &b) const {
        return a > b; // 48簣簣檫髮檫
```

```
    }
};

map<string, int, Compare> myMap;
myMap["Alice"] = 30;
myMap["Bob"] = 25;
for (auto &entry : myMap) {
    cout << entry.first << ":" << entry.second << endl; // 48擦        : Bob
}
```