

# Multiset and Multimap

# Multiset Example

```
#include <set>                                     // Header file for std::multiset

multiset<int> s;                                     // Create an instance of std::multiset
s.insert(6);                                         // Add some elements to it
s.insert(7);
s.insert(4);
s.insert(6);                                         // Add duplicate elements
s.insert(6);

for (auto el: s)                                    // Use a range-for loop
    cout << el << ", ";
```

## Multimap Example

```
#include <map>                                     // Header file for std::multimap

multimap<string, int> scores;                        // Create an instance of std::multimap
scores.insert( {"Graham", 78} );                    // Add some elements to it
scores.insert( {"Grace", 66} );
scores.insert( {"Graham", 66} );                    // Add duplicate elements
scores.insert( {"Graham", 72} );
scores.insert( {"Hareesh", 77} );

for (auto s: scores)                                // Use a range-for loop
    cout << s.first << " has a score of " << s.second << endl;
```

## insert() and erase()

- insert() always succeeds for multimap and multiset
- erase(k) will remove all the elements which have the key k

```
scores.erase("Graham");  
for (auto s: scores) {  
    cout << s.first << " has a score of " << s.second << endl;  
}
```

- We can erase a single element, by passing an iterator to it
- But how do we find it?

## Finding Elements Example

```
auto res = scores.find("Graham");           // Find first element with key "Graham"

if (res != scores.end()) {                  // Do we have any results?
    auto n_matches = scores.count("Graham"); // Find number of matching elements

    // Loop over the matching elements
    for (int i = 0; i < n_matches; ++i) {
        cout << "Key = " << res->first << ", value = " res->second << endl;
        ++res;
    }
}
```

## Finding Elements Example Contd

```
// Sometimes we want to find if there is an element with a given key and a given value
auto res = scores.find("Graham"); // Find first element with key "Graham"

if (res != scores.end()) { // Do we have any results?
    auto n_matches = scores.count("Graham"); // Find number of matching elements

    for (int i = 0; i < n_matches; ++i) { // Loop over the matching elements
        if (res->second == 66) {
            cout << "Found an element with key Graham and value 66!" << endl;
            break;
        }
        ++res;
    }
}
```