C++ Set,MultiSet && UnorderedSet

# Set

## Declaration

set<int> set1;

## Initialization

set<int> set1 = { 10,40,10,20,50,90};

## Iteration && Check

set<int> set1 = { 10,40,10,20,50,90};

for (set<int>::iterator i = set1.begin(); i != set1.end(); i++)

{

cout << \*i << " ";

}

cout << endl;

## Insert

//insert

set1.insert(66);

set1.insert(66);

set1.insert(69);

# MultiSet

## Declaration

multiset<int>set2;

## Initialization

multiset<int>set3 = { 1,3,2,1,4,5,6,7,1,2 };

## Iteration && Check

for (multiset<int>::iterator i = set3.begin(); i != set3.end(); i++)

{

cout << \*i << " ";

}

cout << endl;

## Insert

set3.insert(9);

## Erase

* Iterator:

multiset<int>::iterator del = set3.begin();

set3.erase(del);

* Value: set3.erase(2);
* delete in range: from staert to first aparation of 5:

set3.erase(set3.begin(), set3.find(5));

## Swap

set2.swap(set3);

## Size

cout << "Size of set2 is : " << set2.size()<<endl;

# Unordered map

## Declaration

unordered\_set<int>u\_set1;

## Initialization

unordered\_set<int>u\_set2 = { 10,40,22,333,8,10 };

## Iteration && Check

for(unordered\_set<int>::iterator i = u\_set2.begin();i !=u\_set2.end();i++)

{

cout << \*i << " ";

}

## Insert

u\_set2.insert(9);

## Erase

* Iterator:

unordered\_set<int>::iterator udel = u\_set2.begin();

u\_set2.erase(udel);

* Value: u\_set2.erase(8);
* Delete in range: from staert to first aparation of 22:

u\_set2.erase(u\_set2.begin(), u\_set2.find(22));

## Swap

u\_set1.swap(u\_set2);

## Size

cout << "Size of set2 is : " << u\_set1.size() << endl;