

7COM1025

Programming for Software Engineers

Lecture 18

sort

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
void shows(int i) {
    cout << ' ' << i;
}
int main () {
    vector<int> my_vector;
    int i;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_vector.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    sort(my_vector.begin(), my_vector.end());
    for_each (my_vector.begin(), my_vector.end(), shows);
    return 0;
}
```

UNIQUE

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
void shows(int i) {
    cout << ' ' << i;}
int main () {
    int i;
    vector<int> my_vector;
    vector<int>::iterator ite;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_vector.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    ite=unique(my_vector.begin(), my_vector.end());
    my_vector.resize(ite - my_vector.begin());
    for_each (my_vector.begin(), my_vector.end(), shows);
    return 0;
}
```

Unique in a serialised way.

If input is: 1 2 3 4 4 4 1

Output: 1 2 3 4 1

PROBLEM 18.1

Write a function that receives a vector (integer) and remove ALL duplicated values.

If the input is: 1 2 3 4 4 4 1

The output should be: 1 2 3 4

Your application should show all the elements of the vector before and after using your function.

FOR_EACH

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
void showsquares(int i) {
    cout << ' ' << i*i;
}
```

- Applies a function to each element of a container.
- The range is [first, last).

```
int main () {
    vector<int> my_vector;
    int i;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i)
    {
        my_vector.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    cout << "The square of the elements in my_vector are: ";
    for_each (my_vector.begin(), my_vector.end(), showsquares);
    cout << '\n';
    return 0;
}
```

COUNT

```
#include <iostream>
#include <algorithm>
#include <list>
using namespace std;
void shows(int i) {
    cout << ' ' << i;
}
int main () {
    list<int> my_list;
    list<int>::iterator iter;
    int i, count_i;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_list.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    cin.clear();//resets the fail
    cin.ignore();//removes any wrong input
    cout<<"Enter number to look for: "<<endl;
    cin>>i;
    cout<<i<<" appears "<<count(my_list.begin(), my_list.end(), i)<<" times in the list: "<<endl;
    for_each (my_list.begin(), my_list.end(), shows);
    return 0;
}
```

-Returns the number of elements in [first,last) that evaluates equal to the parameter. It uses the operator ==

count_if

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
int max_i;
bool higherthanmax(int i)
    {return i>max_i;}
void shows(int i)
    {cout << ' ' << i;}
int main () {
    int i;
    vector<int> my_vector;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_vector.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    cin.clear();
    cin.ignore();
    cout<<"Enter desired maximum: ";
    cin>>max_i;
    cout<<"There are "<<count_if(my_vector.begin(), my_vector.end(), higherthanmax)<<" larger than "<<max_i<<endl;
    for_each (my_vector.begin(), my_vector.end(), shows);
    return 0;
}
```

PROBLEM 18.2

Write a program that allows users to enter as many marks (0-100 each) of students as the user wishes.

Write functions for the below:

- 1) One returning the number of students with 70 or more.
- 2) Write a function receiving one numerical parameter which prints its value on the screen.
- 3) Using the previous function to show the marks of all students.

Use STL algorithms whenever appropriate.

FIND

```
#include <iostream>
#include <algorithm>
#include <list>
using namespace std;
void shows(int i) {
    cout << ' ' << i;
}
int main () {
    list<int> my_list;
    list<int>::iterator iter;
    int i;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_list.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    cin.clear();//resets the fail
    cin.ignore();//removes any wrong input
    cout<<"Enter number to look for: "<<endl;
    cin>>i;
    iter = find(my_list.begin(), my_list.end(), i);
    if (iter!=my_list.end())
        cout<<"Element "<<i<<" is in list."<<endl;
    else
        cout<<"Element "<<i<<" is NOT in the list."<<endl;
    cout << "my_list contains: ";
    for_each (my_list.begin(), my_list.end(), shows);
    return 0;
}
```

Returns an iterator to the 1st element in [first, last)

MIN_ELEMENT AND MAX_ELEMENT

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
void shows(int i) {
    cout << ' ' << i;
}
int main () {
    vector<int> my_vector;
    int i;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_vector.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    cout<<"The smallest element is: "<< *min_element(my_vector.begin(), my_vector.end())<<endl;
    cout<<"The highest element is: "<<*max_element(my_vector.begin(), my_vector.end())<<endl;
    return 0;
}
```

Return an iterator

RANDOM_SHUFFLE

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <cstdlib> //for rand
using namespace std;
void shows(int i) {
    cout << ' ' << i;
}
int main () {
    vector<int> my_vector;
    int i;
    cout<<"Enter an integer (enter non-number to stop): "<<endl;
    while (cin>>i){
        my_vector.push_back(i);
        cout<<"Enter an integer (enter non-number to stop): "<<endl;
    }
    for_each (my_vector.begin(), my_vector.end(), shows);
    random_shuffle(my_vector.begin(), my_vector.end());
    cout<<endl;
    for_each (my_vector.begin(), my_vector.end(), shows);
    return 0;
}
```