CSC2111 Computer Science I Lab

Lab 21

Objective

1. Learn how to use vector as a self grown container.

Defining a New Vector

Syntax: `vector<of what>`

For example :

`vector<int>` - vector of integers.
`vector<string>` - vector of strings.
`vector<int * >` - vector of pointers to integers.
`vector<Shape>` - vector of Shape objects. Shape is a user defined class.

Operations on vector

```
iterator begin();
iterator end();
bool empty();
void push_back(const T& x);
iterator erase(iterator it);
iterator erase(iterator first, iterator last);
void clear();
….
```

Example 1

Write a program that read integers from the user, sorts them, and print the result.

```cpp
#include <iostream>  // I/O
#include <vector>    // container
#include <algorithm> // sorting

using namespace std;

int main() {
    int input;
    vector<int> ivec;

    // input
    cout<< "Enter positive number, provide -1 to exit."<< endl;
    cin >> input;
    while(input != -1){
        ivec.push_back(input);
        cin >> input;
    }
    // sorting
    sort(ivec.begin(), ivec.end());

    // output
    cout<< "After sorting: ";
    vector<int>::iterator it;
    for ( it = ivec.begin();
          it != ivec.end(); ++it ) {
            cout << *it << " ";
    }
    cout << endl;
    return 0;
}
```

```
Enter positive number, provide -1 to exit.
4 5 6 7 2 77 55 33 -1
After sorting: 2 4 5 6 7 33 55 77


...Program finished with exit code 0
Press ENTER to exit console.
```

Example 2

Write a program that reads student id and name from the user, and then store them into a vector as a student object. Finally, print the list of all students.

```cpp
#include <iostream>
#include <vector>
#include <string>
using namespace std;

class Student{
private:
   int id;
   string name;
public:
   Student() {id=0; name="";}
   Student(int i, string n){ id = i; name = n;}
   void show() { cout << "Id: "<< id <<", Name: "<< name << endl; }
};

int main()
{
    int stdId;
    string stdName;

    vector<Student> stdVector;
    cout << "Initial size: " << stdVector.size() << endl << endl;

    // input
    cout << "Enter student id and name, provide -1 to exit."<< endl;
    cout << "Enter student id: ";
    cin >> stdId;
    while(stdId != -1){
        cout << "Enter student name: ";
        cin >> stdName;
        stdVector.push_back(Student(stdId, stdName));
        cout << "Enter student id: ";
        cin >> stdId;
    }

    // output
    cout << endl << "List of students:" << endl;
    vector<Student>::iterator it;
    for ( it = stdVector.begin(); it != stdVector.end(); ++it ) {
        (*it).show();
    }
    cout << endl;
    cout << "New size: " << stdVector.size() << endl;
    return 0;
}
```

Output

```
Initial size: 0

Enter student id and name, provide -1 to exit.
Enter student id: 2
Enter student name: Mehedi
Enter student id: 3
Enter student name: Adib
Enter student id: 4
Enter student name: Bob
Enter student id: 7
Enter student name: Adam
Enter student id: -1

List of students:
Id: 2, Name: Mehedi
Id: 3, Name: Adib
Id: 4, Name: Bob
Id: 7, Name: Adam

New size: 4
```