**Learning Objectives**

* Vector
* Map
* Iterator

**Vector**

Vectors are the same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container. Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators. In vectors, data is inserted at the end. Inserting at the end takes differential time, as sometimes the array may need to be extended.

// C++ program to illustrate the

// Modifiers in vector

#include <iostream>

#include <vector>

using namespace std;

int main()

{

// Assign vector

vector<int> v;

// fill the array with 10 five times

v.assign(5, 10);

cout << "The vector elements are: ";

for (int i = 0; i < v.size(); i++)

cout << v[i] << " ";

// inserts 15 to the last position

v.push\_back(15); // add elements 1 by 1 at the back

int n = v.size(); // how big the array

cout << "\nThe last element is: " << v[n - 1];

// removes last element

v.pop\_back();

// prints the vector

cout << "\nThe vector elements are: ";

for (int i = 0; i < v.size(); i++)

cout << v[i] << " ";

// inserts 5 at the beginning

v.insert(v.begin(), 5);

cout << "\nThe first element is: " << v[0];

// removes the first element

v.erase(v.begin());

cout << "\nThe first element is: " << v[0];

// inserts at the beginning

v.emplace(v.begin(), 5);

cout << "\nThe first element is: " << v[0];

// Inserts 20 at the end

v.emplace\_back(20);

n = v.size();

cout << "\nThe last element is: " << v[n - 1];

// erases the vector

v.clear();

cout << "\nVector size after erase(): " << v.size();

// two vector to perform swap

vector<int> v1, v2;

v1.push\_back(1);

v1.push\_back(2);

v2.push\_back(3);

v2.push\_back(4);

cout << "\n\nVector 1: ";

for (int i = 0; i < v1.size(); i++)

cout << v1[i] << " ";

cout << "\nVector 2: ";

for (int i = 0; i < v2.size(); i++)

cout << v2[i] << " ";

// Swaps v1 and v2

v1.swap(v2);

cout << "\nAfter Swap \nVector 1: ";

for (int i = 0; i < v1.size(); i++)

cout << v1[i] << " ";

cout << "\nVector 2: ";

for (int i = 0; i < v2.size(); i++)

cout << v2[i] << " ";

}

**The output is**

**The vector elements are: 10 10 10 10 10**

**The last element is: 15**

**The vector elements are: 10 10 10 10 10**

**The first element is: 5**

**The first element is: 10**

**The first element is: 5**

**The last element is: 20**

**Vector size after erase(): 0**

**Vector 1: 1 2**

**Vector 2: 3 4**

**After Swap**

**Vector 1: 3 4**

**Vector 2: 1 2**

**Iterator**

Iterators are used to point at the memory addresses of STL containers. They are primarily used in sequences of numbers, characters etc. They reduce the complexity and execution time of the program.

// C++ code to demonstrate the working of

// iterator, begin() and end()

#include<iostream>

#include<iterator> // for iterators

#include<vector> // for vectors

using namespace std;

int main()

{

vector<int> ar = { 1, 2, 3, 4, 5 };

// Declaring iterator to a vector

vector<int>::iterator ptr;

// Displaying vector elements using begin() and end()

cout << "The vector elements are : ";

for (ptr = ar.begin(); ptr < ar.end(); ptr++)

cout << \*ptr << " ";

return 0;

}

**Maps**

Maps are a part of the C++ STL. Maps are associative containers that store elements in a combination of key values and mapped values that follow a specific order. No two mapped values can have the same key values.

In-Built functions

Begin(): Returns an iterator to the first element in the map.

size(): Returns the number of elements in the map.

empty(): Returns a boolean value indicating whether the map is empty.

insert( pair(key, value)): Adds a new key-value pair to the map.

find(val): Gives the iterator to the element val, if it is found otherwise it returns m.end()

erase(iterator position): Removes the element at the position pointed by the iterator.

erase(const g): Removes the key value g from the map.

clear(): Removes all the elements from the map.

**Sample of using map**

#include <string>

#include <iostream>

#include <map>

#include <utility>

using namespace std;

int main()

{

// Initializing a map with integer keys

// and corresponding string values

map<int, string> Employees;

//Inserting values in map using insert function

Employees.insert (pair<int, string>(101,"Jon") );

Employees.insert (pair<int, string>(103,"Daenerys") );

Employees.insert (pair<int, string>(104,"Arya") );

// Inserting values using Array index notation

Employees[105] = "Sansa"; // is the same

Employees[102] = "Tyrion";

cout << "Size of the map is " << Employees.size() << endl << endl;

// Printing values in the map

cout << endl << "Default Order of value in map:" << endl;

for( map<int,string>::iterator iter=Employees.begin(); iter!=Employees.end(); ++iter)

{

cout << (\*iter).first << ": " << (\*iter).second << endl;

}

// Finding the value corresponding to the key '102'

std::map<int, string>::iterator it = Employees.find(102);

if (it != Employees.end())

{

std::cout <<endl<< "Value of key = 102 => " << Employees.find(102)->second << '\n';

}

}