CLASSMATE
Date :
Page :

* Object Oriented Programming (OOP) - Practical Number 8 (Group - C)

Name:- Kaustubh Shrikant Kabra.
Class:- Second Year Engineering.
Div:- A Roll Number:-
Batch:-
Department:- Computer Department.
College:- AISSMS's IOIT.

Title:-
Demonstration of implementation of map associative container.

Objective:-
1) To learn and understand concepts of Standard Template Libary.
2) To demonstrate STL for implementation of map associative containes.

Problem Statement:-
Write a program in C++ to use map associative containers. The keys will be the names of states and the value will be the populations of the states. When the program runs, the user is prompted to type the name of state. The program than looks in the map, using the state name as an index and returns the population of the state.

Outcomes:-
1) Student will be able to learn and understand concepts of STL.
2) Student will be able to ~~use~~ implementation map associative containers concepts.

Date :
Page :

Theory:-
Associative containers are those that provide direct access to its element for storage and retrieval purposes. The elements are accessed via keys, also know as search keys. There are four ordered and four unordered associative containers in C++ such as multiset, set, multimap, map and unordered_ multiset, unordered _ set, unordered _ multimap and unordered _ m

• Map Associative Containers:-
The map associative container stores elements as key-value pairs. It uses unique keys to perform fast storage and retrieval of its associative values.
Elements can be inserted and removed from anywhere in the map. I we do not want the constraint of ordering the keys, we can use its unordered version, called the unordered _ map.

Example-
```
#include <iostream>
#include< map >
#include< iterator >
#include < string >
using namespace std;

int main()
{
    map< int, string, less< int>> weekdays;

    weekdays.insert(make_pair(1, "Sunday"));
    weekdays.insert(make_pair(2, "Monday"));
    weekdays.insert(make_pair(4, "Wednesday"));
    weekdays.insert(make_pair(5, "Thursday"));
weekdays. insert(make_pair(7, "Saturday"));
weekdays. insert(make_pair(5, "Thursday"));
weekdays. insert(make_pair(3, "Tuesday"));
weekdays. insert(make_pair(6, "Friday"));

for(auto day: weekays)
   cout << day. first << "_" << day. second << endl;

cout << "\n--------------------------" << endl
weekdays[2] = "Monday";

for(auto day: weekdays)
   cout << day first << "_" << day. second << endl.

return 0;
}.
```

• Multimap Associative Container -

Similar to the map, the multimap associative container is also an associative container. The elements of multimap also are stored in key-value pairs. The relationship between key-value pairs, therefore, is of one-to-many. If we do not want the constraint of ordering the keys, we can use its unordered version called the unordered_multimap.

Example -

```
# include < iostream >
# include < map >
# include < iterator >
# include < string >
```

```
using namespace std;

int main()
{
  multimap< int, string, less< int>> box;

  box.insert(make_pair(1, "socks"));
  box.insert(make_pair(3, "T-Shirt"));
  box.insert(make_pair(6, "gloves"));
  box.insert(make_pair(4, "Shirt"));
  box.insert(make_pair(2, "Jacket"));

  for (auto day: box)
     cout << day.first << "_" << day.second << endl;

 cout << "There are" << box.count(1) << "pair of socks" << endl;

 return 0;
}
```

Algorithm:-

Conclusion:-
We have successfully implemented map associative container.