OBJECT ORIENTED PROGRAMMING IN C++
WINTER SEMESTER 2021-22
TUTORIAL-03

Name: M. DHANUSHRAJ
Reg. No: 21BAI10111.

Question:

1. Write a program to implement STL map with below mentioned functions

- **insert**
- **at**
- **find**
- **swap**
- **empty**
- **clear**
- **size**
- **emplace**
- **erase**
- **end**

Program:

```
#include<string.h>                          //Name: M. DHANUSHRAJ
#include<iostream>                           //Reg. No: 21BAI10111
#include<map>
#include<utility>
#include<iterator>
#include <bits/stdc++.h>
using namespace std;
int main()
{
        map<int, string> Fruits;
        Fruits.insert(pair<int, string>(150, "Apple"));
        Fruits.insert(pair<int, string>(30, "Orange"));
        Fruits.insert(pair<int, string>(20, "Watermelon"));
        Fruits.insert(pair<int, string>(40, "Mango"));
        Fruits.insert(pair<int, string>(80, "Pomegranate"));

        cout<<"1. Insert() function:\n"<<endl;

        map<int, string>::iterator itr;
        for (itr = Fruits.begin(); itr != Fruits.end(); ++itr)
        {
                cout << itr->second << " = " << itr->first << endl;
        }

   cout<<"\n2. at() function:\n"<<endl;
   cout<<Fruits.at(20)<<endl;
```

```
    cout<<Fruits.at(40)<<endl;

    cout<<"\n3. find() function:\n"<<endl;
    itr= Fruits.find(80);
    cout<<"Iterator points to "<< itr-> first <<"="<< itr-> second<<endl;

    cout<<"\n4. swap() function:\n"<<endl;

    map<int, string> Vegetables;
    Vegetables.insert(pair<int, string>(10, "Tomato"));
    Vegetables.insert(pair<int, string>(25, "Onion"));
    Vegetables.insert(pair<int, string>(20, "Brinjal"));
    Vegetables.insert(pair<int, string>(15, "Cabbage"));
    Vegetables.insert(pair<int, string>(30, "Potato"));

    Fruits.swap(Vegetables);

    cout<<"\nAfter swaping, Fruits contains:\n";
    for(itr=Fruits.begin(); itr!=Fruits.end(); itr++)
    {
        cout<<endl<<itr->first<<" = "<<itr->second;
        }

        cout<<"\n5. empty() function:\n"<<endl;
        cout<<"Fruits is Empty or not:"<<Fruits.empty()<<endl;
        cout<<"Vegetables is Empty or not:"<<Vegetables.empty()<<endl;

        cout<<"\n6. clear() function:\n"<<endl;
        Vegetables.clear();
        //No values will be there after clear() function.

        cout<<"\n7. size() function:\n"<<endl;
        cout<<"Size of Vegetables:"<<Vegetables.size()<<endl;
        cout<<"Size of Fruits:"<<Fruits.size()<<endl;

        cout<<"\n9. erase() function:"<<endl;
        itr=Fruits.find(30);
        Fruits.erase(itr);
        cout<<"After erasing, the Fruits contains:\n:";
        for(itr=Fruits.begin(); itr!=Fruits.end(); itr++)
    {
        cout<<endl<<itr->first<<" = "<<itr->second;
        }

        cout<<"\n\n10. end() function:";
        //end() function is used to return an iterator pointing next to the last element
        //It is already used so many times in the code.
    return 0;
}
```

OUTPUT:

```
C:\Users\Dhaush Raj\Documents\stl map.exe
1. Insert() function:

Watermelon = 20
Orange = 30
Mango = 40
Pomegranate = 80
Apple = 150

2. at() function:

Watermelon
Mango

3. find() function:

Iterator points to 80=Pomegranate

4. swap() function:


After swaping, Fruits contains:

10 = Tomato
15 = Cabbage
20 = Brinjal
25 = Onion
30 = Potato
5. empty() function:

Fruits is Empty or not:0
Vegetables is Empty or not:0

6. clear() function:


7. size() function:

Size of Vegetables:0
Size of Fruits:5

9. erase() function:
After erasing, the Fruits contains:
:
10 = Tomato
15 = Cabbage
20 = Brinjal
25 = Onion

10. end() function:
--------------------------------
Process exited after 5.311 seconds with return value 0
Press any key to continue . . .
```