Institute of Computer Technology

B. Tech Computer Science and Engineering

Subject: ESFP-II (2CSE203)

**PRACTICAL-14**

**AIM: - To learn about STL in C++.**

**Q.1. Mr. Omprakash want to accept n numbers of element from user in arraylist, and at the same he wants to show all the element of arraylist in list. So, write the appropriate C++ program for achieve the concept of arraylist and list through STL.**

**Input : 1,2,3,4,5,6 [ in arraylist]**

**Output : 1,2,3,4,5,6 [ in list ]**

***CODE:***

#include<iostream>

#include<list>

using namespace std;

int main()

{

list <int> ob;

int arr[20];

int n;

int \*p;

cout<<"\nEnter number of data you want to enter: ";

cin>>n;

for (int i = 0; i < n; i++)

{

cout<<"\nEnter "<<i+1<<" data: ";

cin>>arr[i];

}

for (int i = 0; i < n; i++)

{

p=&arr[i];

ob.push\_back(\*p);

}

cout<<"\nArrayList elements: "<<endl;

for (int i = 0; i < n; i++)

{

cout<<arr[i]<<"\t";

}

list <int> ::iterator it;

it=ob.begin();

cout<<"\nList elements: "<<endl;

for (;it!=ob.end();it++)

{

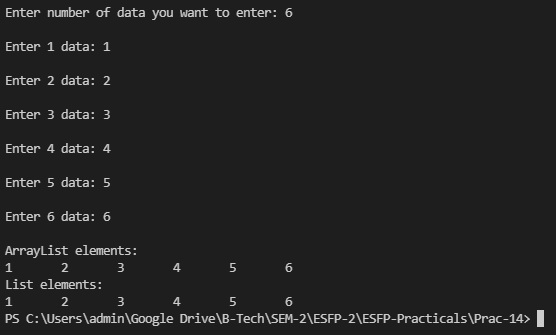
cout<<\*it<<"\t";

}

return 0;

}

***OUTPUT:***

******

**Q.2. Mr.Rupesh Malhotra, want to perform some operations through STL in C++ program, In vector he stores 10 number from user. Now, perform following operation in vector with the help of below given STL function.**

**size(): It will shows numbers of elements.**

**push\_back() : Append / add an element to the end of list.**

**pop\_back() : Erase the last element.**

**begin() : Provide reference to last element.**

**end() : Provide reference to end of vector.**

***CODE:***

#include<iostream>

#include<vector>

using namespace std;

int main()

{

vector <int> obj;

int x;

for (int i = 0; i < 10; i++)

{

cout<<"Enter data "<<i+1<<" : ";

cin>>x;

obj.push\_back(x);

}

cout<<"\nSize of vector: "<<obj.size()<<endl;

cout<<"\nAppending element to the end of list: "<<endl;

obj.push\_back(40);

for (vector<int>::iterator it = obj.begin() ; it != obj.end(); ++it)

{

cout<< \*it<<"\t";

}

cout<<"\nRemove last element from vector: "<<endl;

obj.pop\_back();

for (vector<int>::iterator it = obj.begin() ; it != obj.end(); ++it)

{

cout<< \*it<<"\t";

}

cout<<"\nEnter number at the beginning of vector list: "<<endl;

vector <int> ::iterator it2;

it2=obj.begin();

obj.insert(it2,900);

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

{

cout<<obj[i]<<"\t";

}

cout<<"\nEnter number at the end of vector list: "<<endl;

vector <int> ::iterator it3;

it3=obj.end();

obj.insert(it3,1100);

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

{

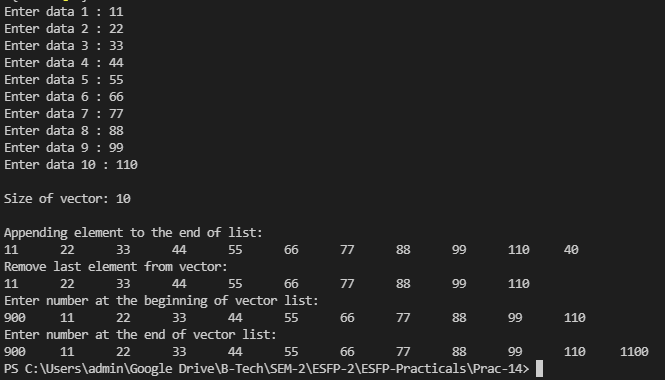
cout<<obj[i]<<"\t";

}

return 0;

}

***OUTPUT:***

****

***Post Practical Task***

**1. Make a C++ program, where you have to perform stack and queue concept through STL.**

**[ Follow the following instruction]**

1. **Insert element in stack and Queue**
2. **Append element in Stack and Queue**
3. **Delete element from stack and Queue.**

***CODE:***

#include <iostream>

#include <stack>

using namespace std;

int main()

{

int n,m;

stack <int> obj;

cout<<"\nInserting values in stack.";

cout<<"\nDisplaying values in stack: "<<endl;

obj.push(10);

obj.push(20);

obj.push(30);

obj.push(40);

n=obj.size();

for (int i = 0; i < n; i++)

{

cout<<obj.top()<<"\t";

obj.pop();

}

cout<<"\nAppending values in stack: "<<endl;

obj.push(10);

obj.push(20);

obj.push(30);

obj.push(40);

obj.push(50);

n=obj.size();

for (int i = 0; i < n; i++)

{

cout<<obj.top()<<"\t";

obj.pop();

}

cout<<"\nDeleting 2 values from stack: "<<endl;

obj.push(10);

obj.push(20);

obj.push(30);

obj.push(40);

obj.push(50);

obj.pop();

obj.pop();

m=obj.size();

for (int i = 0; i < m; i++)

{

cout<<obj.top()<<"\t";

obj.pop();

}

return 0;

}

***OUTPUT:***

******

**2. Make a program in C++, where you have to insert element in the format of key and value pair in map. Perform the following below given operation.**

1. **insert element**
2. **modify element**
3. **delete element**

***CODE:***

#include<iostream>

#include<map>

using namespace std;

int main()

{

map <int,int> m;

m.insert(pair<int, int>(1, 100));

m.insert(pair<int, int>(2, 200));

m.insert(pair<int, int>(3, 300));

m.insert(pair<int, int>(4, 400));

cout<<"Displaying All elements:"<<endl;

map <int,int> :: iterator it;

for (it = m.begin(); it != m.end(); it++)

{

cout << "key ID:" << it->first << "\tValue:" << it->second << endl;

}

cout<<"Modified Elements:"<<endl;

map <int,int> :: iterator it1;

it1=m.begin();

m.insert(pair<int,int>(5,1000));

for (it1 = m.begin(); it1 != m.end(); it1++)

{

cout << "key ID:" << it1->first << "\tValue:" << it1->second << endl;

}

cout<<"Deleted single element(3rd element):"<<endl;

m.erase(3);

map <int,int> :: iterator it2;

for (it2 = m.begin(); it2 != m.end(); it2++)

{

cout << "key ID:" << it2->first << "\tValue:" << it2->second << endl;

}

cout<<"All Elements deleted"<<endl;

m.clear();

map<int, int>::iterator it3;

for (it3 = m.begin(); it3 != m.end(); it3++)

{

cout << "key ID:" << it3->first << "\tValue:" << it3->second << endl;

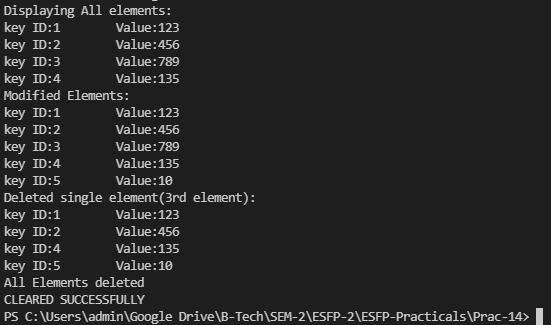
}

cout<<"CLEARED SUCCESSFULLY";

return 0;

}

***OUTPUT:***

****

**Q.3. find the output:**

**#include <vector>**

**#include <algorithm>**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**vector<int> v(10, 2);**

**if (all\_of(v.cbegin(), v.cend(), [](int i){ return i % 2 == 0; }))**

**{**

**cout << "Even Number";**

**}**

**else**

**{**

**cout << "Odd Number";**

**}**

**return 0;**

**}**

***a) Even Number***

b) All numbers are not even

c) Error

d) Segmentation fault

***OUTPUT:***

******

**4. find the output of program.**

**#include <vector>**

**#include <algorithm>**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**vector<int> v;**

**for(int i=0;i<10;i++)**

**v.push\_back(i+1);**

**for(int i=0;i<10;i++)**

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

**cout<<endl;**

**random\_shuffle(v.begin(), v.end());**

**for(int i=0;i<10;i++)**

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

**return 0;**

**}**

a. 1 2 3 4 5 6 7 8 9 10

5 4 8 9 1 6 3 2 7 10

b. 1 2 3 4 6 7 8 9 10 11

5 6 7 8 9 3 2 1 4 10

***c. 1 2 3 4 5 6 7 8 9 10***

***9 2 10 3 1 6 8 4 5 7***

d. None of the above.

***OUTPUT:***

****