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:

```
Enter number of data you want to enter: 6

Enter 1 data: 1

Enter 2 data: 2

Enter 3 data: 3

Enter 4 data: 4

Enter 5 data: 5

Enter 6 data: 6

ArrayList elements:
1       2       3       4       5       6
List elements:
1       2       3       4       5       6
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-14>
```

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:

```
Enter data 1 : 11
Enter data 2 : 22
Enter data 3 : 33
Enter data 4 : 44
Enter data 5 : 55
Enter data 6 : 66
Enter data 7 : 77
Enter data 8 : 88
Enter data 9 : 99
Enter data 10 : 110

Size of vector: 10

Appending element to the end of list:
11      22      33      44      55      66      77      88      99      110     40
Remove last element from vector:
11      22      33      44      55      66      77      88      99      110
Enter number at the beginning of vector list:
900     11      22      33      44      55      66      77      88      99      110
Enter number at the end of vector list:
900     11      22      33      44      55      66      77      88      99      110     1100
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-14>
```

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:

```
/*
   Stack is a container adaptor.
   Stack follows rules of LIFO (last in first out)
*/
#include<iostream>
#include<stack>
#include<queue>
using namespace std;
int main()
{
   int size,num;
   stack <int> s;
```

```
queue <int> q;
cout<<"\n===============| Stack Example |===============\n";
cout<<"Enter the size of stack:";
cin>>size;
cout<<"\nEnter the element of stack:";
for(int i=0;i<size;i++)
{
   cin>>num;
   s.push(num);
}

cout<<"\nSize of stack:"<<s.size()<<endl;
cout<<"\nOutput of stack element:\n";
while(!s.empty())
{
   cout<<s.top()<<"\t";
   s.pop();
}
cout<<"\nSize of stack:"<<s.size()<<endl;
cout<<"\nInput element in stack:\n";
s.push(999);
s.push(888);
s.push(777);
s.push(666);
while(!s.empty())
{
   cout<<s.top()<<"\t";
   s.pop();
}
cout<<"\nDeleting 2 values from stack: "<<endl;
s.push(999);
s.push(888);
s.push(777);
s.push(666);
s.pop();
s.pop();
while(!s.empty())
{
   cout<<s.top()<<"\t";
   s.pop();
}
```

```
    cout<<"\n===============| Queue Example |===============\n";
    cout<<"Enter the size of queue:";
    cin>>size;
    cout<<"\nEnter the element of queue:";
    for(int i=0;i<size;i++)
    {
       cin>>num;
       q.push(num);
    }

    cout<<"\n Size of queue:"<<q.size()<<endl;
    cout<<"\n Output of queue element:\n";
    while(!q.empty())
    {
       cout<<q.front()<<"\t";
       q.pop();
    }
    cout<<"\n Size of queue:"<<q.size()<<endl;
    cout<<"\n Input element in queue:\n";
    q.push(123);
    q.push(456);
    q.push(789);
    while(!q.empty())
    {
       cout<<q.front()<<"\t";
       q.pop();
    }
    cout<<"\nDeleting 2 values from stack: "<<endl;
    q.push(123);
    q.push(456);
    q.push(789);
    q.push(121);
    q.pop();
    q.pop();
    while(!q.empty())
    {
       cout<<q.front()<<"\t";
       q.pop();
    }
    return 0;
}
```

}

OUTPUT:

```
==============| Stack Example |==============
Enter the size of stack:3

Enter the element of stack:10 20 30

Size of stack:3

Output of stack element:
30      20      10
Size of stack:0

Input element in stack:
666     777     888     999
Deleting 2 values from stack:
888     999
==============| Queue Example |==============
Enter the size of queue:3

Enter the element of queue:90 80 70

 Size of queue:3

 Output of queue element:
90      80      70
 Size of queue:0

 Input element in queue:
123     456     789
Deleting 2 values from stack:
789     121
PS C:\Users\Admin\Google Drive\B-Tech\SEM-2\ESFP-2\Lec Notes\Codes>
```

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:

```
Displaying All elements:
key ID:1        Value:123
key ID:2        Value:456
key ID:3        Value:789
key ID:4        Value:135
Modified Elements:
key ID:1        Value:123
key ID:2        Value:456
key ID:3        Value:789
key ID:4        Value:135
key ID:5        Value:10
Deleted single element(3rd element):
key ID:1        Value:123
key ID:2        Value:456
key ID:4        Value:135
key ID:5        Value:10
All Elements deleted
CLEARED SUCCESSFULLY
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-14>
```

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:

```
Even Number
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-14>
```

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:

```
1 2 3 4 5 6 7 8 9 10
9 2 10 3 1 6 8 4 5 7
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-14>
```