Page No.
Date

Anshu Surana
22-23, 1032210755
Batch C1

DCCL Lab Assignment - 5A

Problem statement:
Write a C++ program to perform various operations as below on a list of items using vectors in STL. (ADD, REMOVE, FIND, SORT, PRINT, EXIT)

Objectives:
To implement the concept of STL.

Theory:
Templates:
Templates in C++ is defined as a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using templates.

There are two types of templates:
i) Function template:
A function template allows you to define a function that can accept arguments of different types.

```
template <class T>
T add (Ta, Tb) {
    return a+b;
}
```

Teacher's Sign:

2) class template:

A class template allows you to define a function that can accept work with different types of data

```
template <class T>
class  class-name
{  class members
}
```

STL: The standard Template library is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms and iter. It's a generalized library and so, its components are parameterized.

Different types of Containers:

- Sequence containers: Implement data structures which can be accessed in a sequential manner.
  - vector
  - list
  - Deque.
- Container Adaptors: Provide a different interface for sequential containers.
  - Queue
  - Priority Queue
  - Stack
- Associative Containers: Implement sorted data structures that can be quickly searched
  - set, - multiset, map, multimap

Teacher's Sign.

Platform: 64

Input: List of

Output: Displa

Conclusion: H
STL in C

FAQ's

What is G
Generic pr
emphasizes
components
types and
in which
specified-
for speci

2) What are
-> 1. Reusabi
can be c
2. Efficien
structures
which
to custo
3. Large
which
who co
tutorial

Page No.
Date

Platform: 64-bit open-source Linux

Input: List of items

Output: Display as per the choice specified

Conclusion: Hence, studied how to apply the concepts of STL in c++.

FAQ's

1) What is generic programming?

Generic programming is a programming paradigm that emphasizes the creation of flexible and reusable software components that can be applied to a wide variety of data types and algorithms. It is a type of computer programming in which algorithms are written in terms of types to-be-specified-later that are then instantiated when needed for specific types provided as parameters.

2) What are the advantages of using STL in c++?

-> 1. Reusability: One of the key advantages of the STL is that can be applied to different data types

2. Efficient algorithms: Many of the algorithms and data structures in the STL are implemented using optimized algorithm which can result in faster execution times compared to custom code.

3. Large community of users: The STL is widely used, which means that there is a large community of developers who can provide support and resources, such as tutorials and forums.

Teacher's Sign:

3) What are the different types of iterators?

→ i) Input iterator
ii) Output iterator
iii) Forward iterator
iv) Bidirectional iterator
v) Random access iterator.

Algorithm:

1) Start
2) Import algorithm iterator and vector header files
3) In main function declare a vector of data type int
4) Create switch statement
5) case 1: Add elements using push back
   2: Remove item
   3: find element using find().
   4: sort list using sort().
   5: print the elements using iterators.
6) End.

Ap 22/4/23

Teacher's Sign.

Page No.
Date

Devanshu Saxena
RC-23, 1032210755
Batch C1

OOCCJ Lab Assi

Problem statement

Objective: To stud

Theory:
1) Array List:
import java.util

public class Ar
public static
Array List <s
fruits.ad
fruits.a
fruits.a
fruits.a
System.out.p
fruits.remov
System.out.pr

System.out.p

}

Name: Devanshu Surana

Roll No.: 23

Panel: C

Batch: C1

OOCCJ Lab Assignment 5A

CODE:

```
#include<iostream>
#include <stdio.h>
#include<vector>
#include<algorithm>
using namespace std;

class vectors
{
   public:
   vector <int> v;
   vector <int> v1;
   vector <int> vnew;
   vector <int> :: iterator itr,itr1;
   void pop();
   void push();
   void display();
   void Sort();
   void find1();
};
```

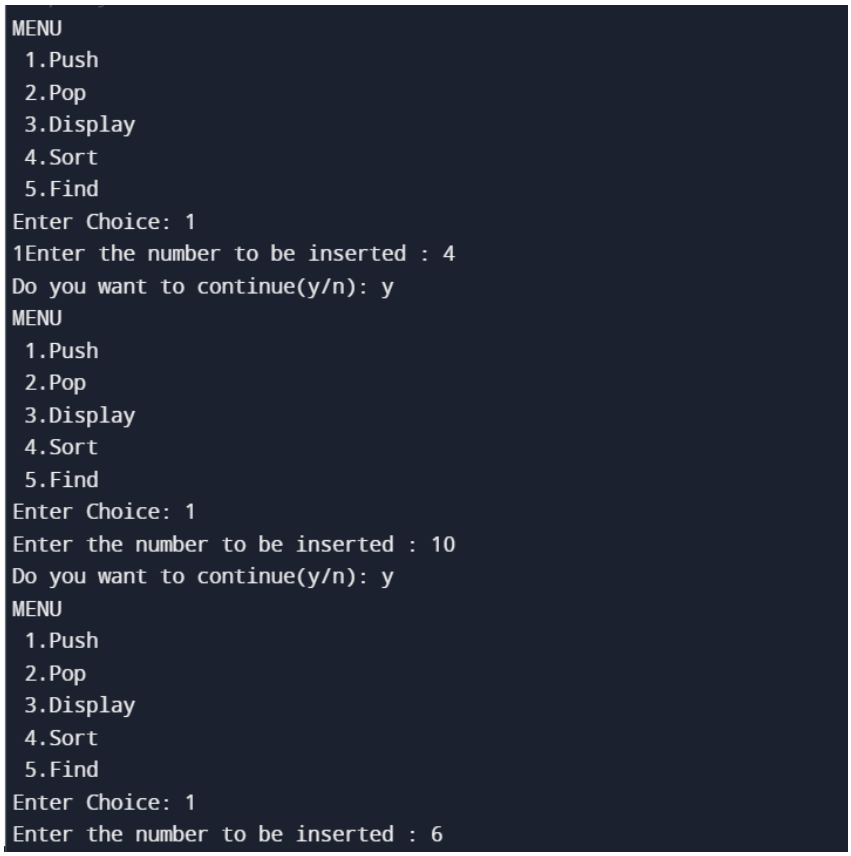
```
void vectors :: pop()
{
   int num;
   itr = v.end();
   itr-- ;
   num =*itr;
   v.pop_back();
   cout << "The element popped is: "<<num<<endl;
}

void vectors :: push()
{
   int num;
   cout << "Enter the number to be inserted : ";
   cin >> num;
   v.push_back(num);
   cout << endl;
}

void vectors :: display()
{
   cout << "The elements in the vector are : ";
   for (itr = v.begin() ; itr!=v.end() ; itr ++)
   {
      cout << *itr << " ";
```

```
    }
    cout << endl;
}

void vectors::Sort()
{
  cout<<"Sorted: ";
    sort(v.begin(), v.end());
    for(itr=v.begin();itr!=v.end();itr++)
    {
        cout << *itr << " ";
    }
}

void vectors::find1()
{
    int ser;
    cout<<"Enter element to be searched: ";
    cin>>ser;
    itr1=find(v.begin(), v.end(), ser);
    if (itr1!= v.end())
      {
        cout << "Element " << ser << " found.";
      }
      else
```

```
        cout << "Element not found.\n\n";
}

int main()
{
    int choice;
    char opt;
    vectors v,v1,vnew;
    do
    {
        cout << "MENU \n 1.Push \n 2.Pop \n 3.Display \n 4.Sort \n 5.Find
\nEnter Choice: ";
        cin >> choice;
        switch(choice)
        {
            case 1:
                v.push();
                 break;
            case 2:
                v.pop();
                break;
            case 3:
                v.display();
                break;
            case 4:
                v.Sort();
```

```
            break;
        case 5:
            v.find1();
            break;
        default:
            cout << "WRONG INPUT"<<endl;
            break;
        }
      cout << "\nDo you want to continue(y/n): ";
      cin >> opt;
   }while(opt == 'y');
}
```

OUTPUT:



```
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 1
1Enter the number to be inserted : 4
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 1
Enter the number to be inserted : 10
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 1
Enter the number to be inserted : 6
```

```
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 1
Enter the number to be inserted : 12
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 1
Enter the number to be inserted : 1
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 3
The elements in the vector are : 4 10 6 12 1
```

```
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 2
The element popped is: 1

Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 3
The elements in the vector are : 4 10 6 12

Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
```

```
Enter Choice: 4
Sorted: 4 6 10 12
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 5
Enter element to be searched: 4
Element 4 found.
Do you want to continue(y/n): y
MENU
 1.Push
 2.Pop
 3.Display
 4.Sort
 5.Find
Enter Choice: 6
WRONG INPUT

Do you want to continue(y/n):
```