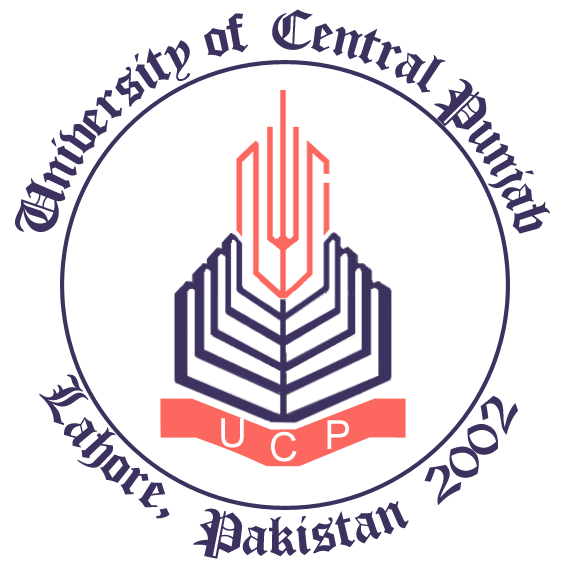
DSA SETS Lab 2



Topic: Sets

Session: Spring 2023

Faculty of Information Technology

UCP Lahore Pakistan

Table of Contents

Topic: Sets

[Objectives 3](#_Toc450131862)[Sample Task 3](#_Toc450131863)[Tasks 5](#_Toc450131864)

## 

## **Objectives**

At the end of the lab the students will have their own class of sets which can perform the functionalities of the sets.

**Sets:**

A set can also be considered as a list of discrete elements, but it differs from a list in which the elements are ordered, i.e. a set holds no duplicates. A major difference between a set and a list is that every element in a set is distinct, i.e. an element cannot exist more than once. For example, *{1, 2}* can be a list or a set, but *{1, 2, 2}* cannot be a set. In a set repetition of a member is not allowed. In fact, every element present in a set forms a representation. For instance, we say that a set of items i.e. *{curtains, furniture, light-fittings}* is required to furnish a house. The set becomes redundant if we say that the required set is *{curtains, curtains, furniture, furniture, light-fittings, light-fittings}*.

A set is merely a collection of elements with no order.

The operations that are applicable to sets include:

 Add an Element to a set.

 Remove an element from set.

 Find out if an element is present in a set

 Find out whether a set is empty.

 Add elements of one set to another set(i.e union operation on sets).

 Find out the common elements between two sets (i.e intersection operation on sets).

 Find out the difference between two sets(i.e difference operation).

## **Sample Task (Add Element, Remove Element, Display)**

#include <iostream>

using namespace std;

class MyArray

{

int\* arr;

int size;

int count;

void init()

{

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

{

arr[i] = -1;

}

}

public:

MyArray(int s = 5)

{

size = s;

arr = new int[size];

count = 0;

init();

}

bool checkBound(int index)

{

if (index<0 && index>size - 1)

return true;

return false;

}

void insertAtEnd(int val)

{

if (!checkBound(count))

{

arr[count] = val;

count++;

}

}

void deleteAtEnd()

{

if (count != 0)

{

arr[count] = -1;

count--;

}

}

void deleteAtIndex(int index)

{

if (!checkBound(index) && arr[index] != -1)

{

arr[index] = arr[count - 1];

arr[--count] = -1;

}

}

int search(int key)

{

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

{

if (arr[i] == key)

{

return i;

}

}

return -1;

}

void display()

{

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

{

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

}

cout << endl;

}

};

class Sets

{

MyArray mySet;

public:

Sets(int s = 5) :mySet(s) {}

void addElement(int val)

{

if (mySet.search(val) == -1)

{

mySet.insertAtEnd(val);

}

else

{

cout << "Value already exists" << endl;

}

}

void removeElement(int val)

{

int index = mySet.search(val);

mySet.deleteAtIndex(index);

}

void display()

{

mySet.display();

}

};

int main()

{

Sets s1;

s1.addElement(10);

s1.addElement(100);

s1.addElement(50);

s1.addElement(90);

s1.display();

s1.removeElement(100);

s1.display();

return 0;

}

## **Tasks**

Your tasks are as follow:

1. Make a new class with the name Mysets, the class should be template type.

You are given the following Set class declaration

template <class Type>

class Set

{

public:

Set();

~Set();

void add\_member(Type elem);

void rm\_member();

intis\_empty();

intis\_full();

bool find(Type elem);

void compute\_union(const Set &obj);

void compute\_intersection(const Set &obj);

void compute\_difference(const Set &obj);

};

Your task is to provide definition of these functions and test them in the main by providing user with an appropriate menu.