**INTEGER SET**

**IMPLEMENTATION**

**LAB # 0****3**

**Fall 2022**

**CSE-208**

**Object Oriented Programming**

Submitted by: **AIMAL KHAN**

Registration No.: **21PWCSE1996**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr. Nasru-min-Allah**

January 19, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Lab no. 3: Integer Set**

**Let us model a class Set. A set contains a number only once. Create a class Set that has an array of 50 members. This set can have values in the range 0-49. If a number is present in Set, the value of array element at that index is 1. Otherwise a 0 is stored.**

1. Write a no-argument constructor that initializes the set to empty (All the array elements are 0).
2. Write a function **insert()** to add a value in the set. (set that index element to 1)
3. Write a function **delete()** to remove a member from the set. (set that index element to 0)
4. Write an overloaded operator + function to find the union of two sets.
5. Write an overloaded operator \* to find the intersection of two sets.
6. Write an overloaded operator ~ to find the complement of a set.
7. Write a display function to display the contents of a set.

**CODE:**

#include <iostream>

using namespace std;

const int MAX = 20;

class IntegerSet {

private:

int st[MAX];

public:

// No-argument constructor that initializes the set to empty

IntegerSet() {

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

st[i] = 0;

}

// Function to add a value in the set

void insertElement(int value) {

st[value] = 1;

}

// Function to remove a member from the set

void deleteValue(int value) {

st[value] = 0;

}

// Overloaded operator + function to find the union of two sets

IntegerSet operator+(const IntegerSet& other) {

IntegerSet result;

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

if (st[i] == 1 || other.st[i] == 1)

result.insertElement(i);

return result;

}

// Overloaded operator \* function to find the intersection of two sets

IntegerSet operator\*(const IntegerSet& other) {

IntegerSet result;

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

if (st[i] == 1 && other.st[i] == 1)

result.insertElement(i);

return result;

}

// Overloaded operator ~ function to find the complement of a set

IntegerSet operator~() {

IntegerSet result;

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

if (st[i] == 0)

result.insertElement(i);

return result;

}

// Function to display the contents of a set

void display() {

cout <<"{ ";

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

if (st[i] == 1)

cout << i << ", ";

cout << "\b\b }\n";

}

};

int main(){

IntegerSet s, t;

s.insertElement(2);

s.insertElement(5);

s.insertElement(7);

s.insertElement(3);

s.insertElement(18);

s.insertElement(16);

cout<< "Set s: ";s.display();

s.deleteValue(2);

s.deleteValue(8);

cout<< "After Deleting 2 and 8, s: ";s.display();

t.insertElement(3);

t.insertElement(5);

t.insertElement(7);

t.insertElement(13);

cout<< "\nSet t: ";t.display();

t.deleteValue(3);

cout<< "After Deleting 3, t: ";t.display();

IntegerSet u = s + t;

cout<< "\nUnion of s and t: ";u.display();

IntegerSet i = s \* t;

cout<< "\nIntersection of s and t: ";i.display();

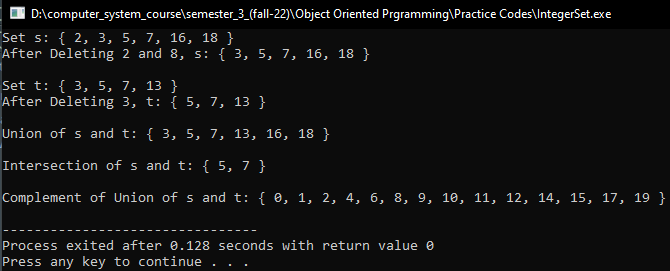
IntegerSet inv = ~u;

cout<< "\nComplement of Union of s and t: ";inv.display();

return 0;

}

**Output:**

****