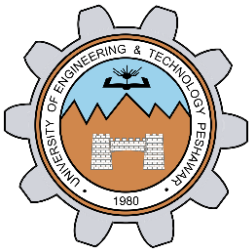
ASSIGNMENT 3



Fall 2022

Object Oriented Programming

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

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

Submitted to:

Dr. Nasruminallah

January 20, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Name: Ali Asghar Section: C Registration # :21PWCSE2059

QUESTION # 1:

CODE SCREENSHOTS:

Here are the screenshots of the code.

main - Copy.cpp | Activity_4_5_1.cpp | main.cpp

```cpp
#include<iostream>
using namespace std;
class Set
{
  private:
    int set[50];

  public:
    Set()
    {
        for (int i = 0; i < 50; i++)
            set[i] = 0;
    }

    void insert(int value)
    {
        if (value >= 0 && value < 50)
            set[value] = 1;
    }

```

Name: Ali Asghar Section: C Registration # :21PWCSE2059

main - Copy.cpp | Activity_4_5_1.cpp | main.cpp

```
20
21      void remove(int value)
22      {
23          if (value >= 0 && value < 50)
24              set[value] = 0;
25      }
26
27      Set operator+(Set &other)
28      {
29          Set unionSet;
30          for (int i = 0; i < 50; i++)
31              unionSet.set[i] = set[i] | other.set[i];
32          return unionSet;
33      }
34
35      Set operator*(Set &other)
36      {
37          Set intersectionSet;
38          for (int i = 0; i < 50; i++)
```

main - Copy.cpp | Activity_4_5_1.cpp | main.cpp

```
37          Set intersectionSet;
38          for (int i = 0; i < 50; i++)
39              intersectionSet.set[i] = set[i] & other.set[i];
40          return intersectionSet;
41      }
42
43      Set operator~()
44      {
45          Set complementSet;
46          for (int i = 0; i < 50; i++)
47              complementSet.set[i] = !set[i];
48          return complementSet;
49      }
50
51      void display()
52      {
53          for (int i = 0; i < 50; i++)
54          {
55              if (set[i] == 1)
```

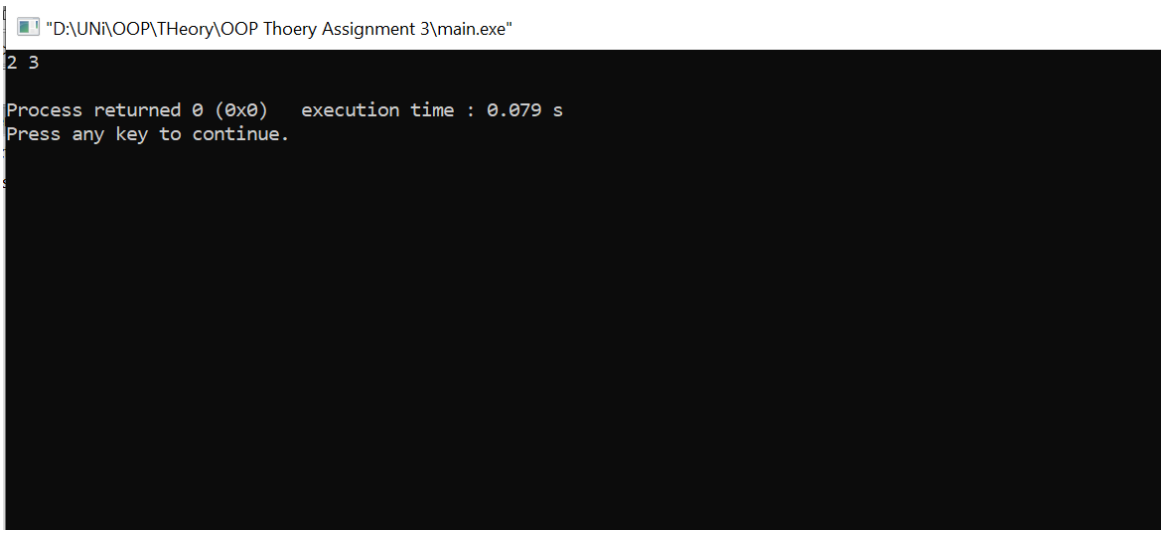
Name: Ali Asghar Section: C Registration # :21PWCSE2059

main - Copy.cpp | Activity_4_5_1.cpp | main.cpp

```
52      {
53          for (int i = 0; i < 50; i++)
54          {
55              if (set[i] == 1)
56                  cout << i << " ";
57          }
58          cout << endl;
59      }
60  };
61
62  int main(){
63
64      Set v;
65      v.insert(2);
66      v.insert(3);
67      v.display();
68      return 0;
69  }
```

OUTPUT (COMPILATION, DEBUGGING & TESTING):

Here is the screenshot of the output of above code.



Name: Ali Asghar Section: C Registration # :21PWCSE2059

IMPLEMENTATION STRATEGY:

The class has a private member variable called "set" which is an array of integers with a size of 50. The class has several member functions that allow for manipulating the set, including:

- The constructor: This function initializes all elements of the "set" array to 0.
- The "insert" function: This function takes an integer value as an input and sets the corresponding element of the "set" array to 1.
- The "remove" function: This function takes an integer value as an input and sets the corresponding element of the "set" array to 0.
- The "+" operator overload: This function takes another "Set" object as an input and returns a new "Set" object that contains all elements that are present in either of the two sets.
- The "*" operator overload: This function takes another "Set" object as an input and returns a new "Set" object that contains all elements that are present in both sets.
- The "~" operator overload: This function returns a new "Set" object that contains all elements that are not present in the original set.
- The "display" function: This function iterates through the "set" array and prints out all elements with a value of 1.

In the main function, an instance of the "Set" class is created and the "insert" function is called twice to add the values 2 and 3 to the set. The "display" function is then called to print out the contents of the set, which should be 2 and 3.