AIR UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EXPERIMENT NO 2

Lab Title Introduction to Classes in C++

Student Name:________________________________**Reg. No:** ____________________

Objective: ____

____

LAB ASSESSMENT:

Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Ability to Conduct Experiment					
Ability to assimilate the results					
Effective use of lab equipment and follows the lab safety rules					

Total Marks: ____________________ Obtained Marks: ____________________

LAB REPORT ASSESSMENT:

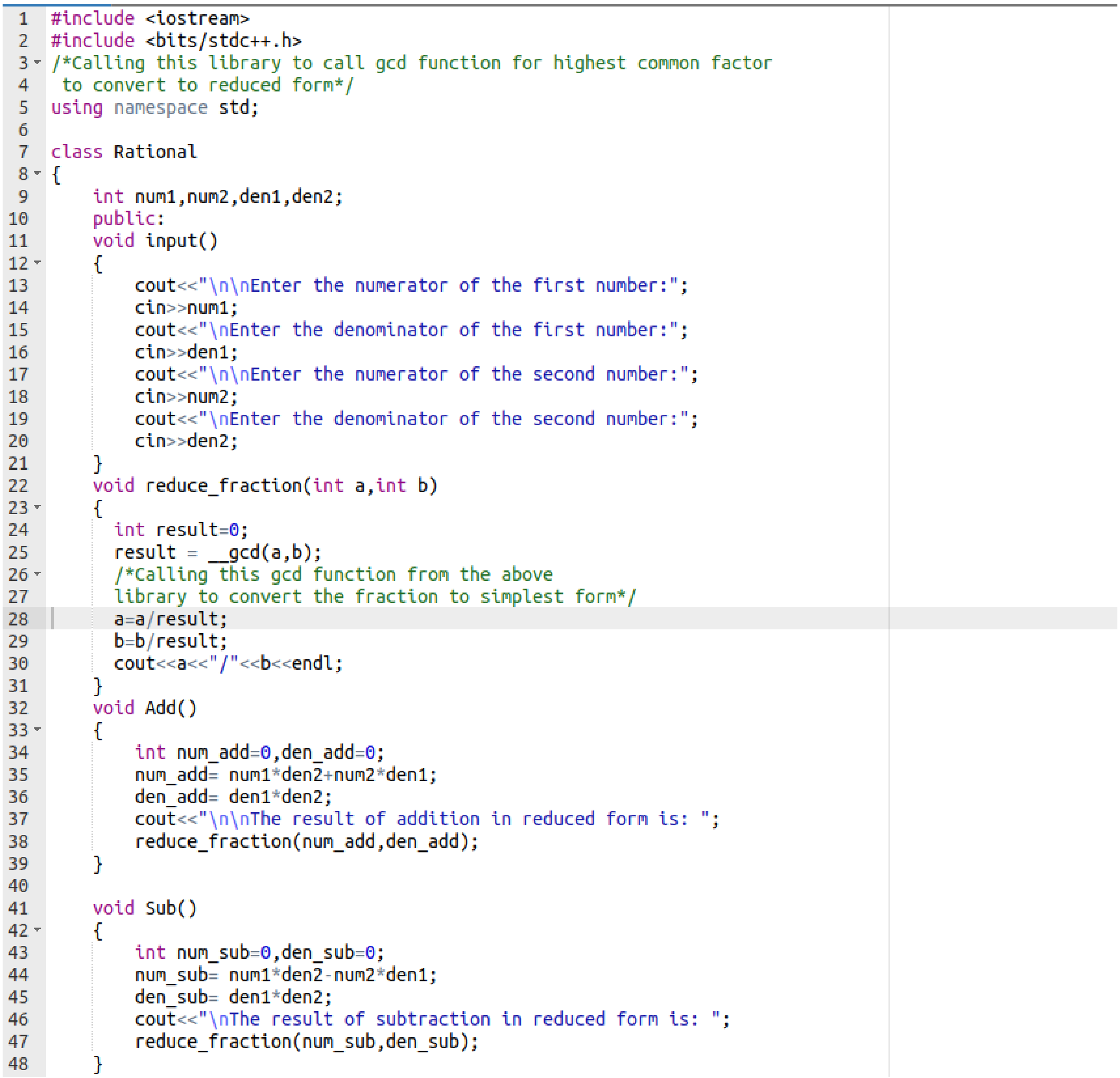
Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Data presentation					
Experimental results					
Conclusion					

Total Marks: ____________________ Obtained Marks: ____________________

Date: ____________________ Signature: ____________________

LAB TASK # 2

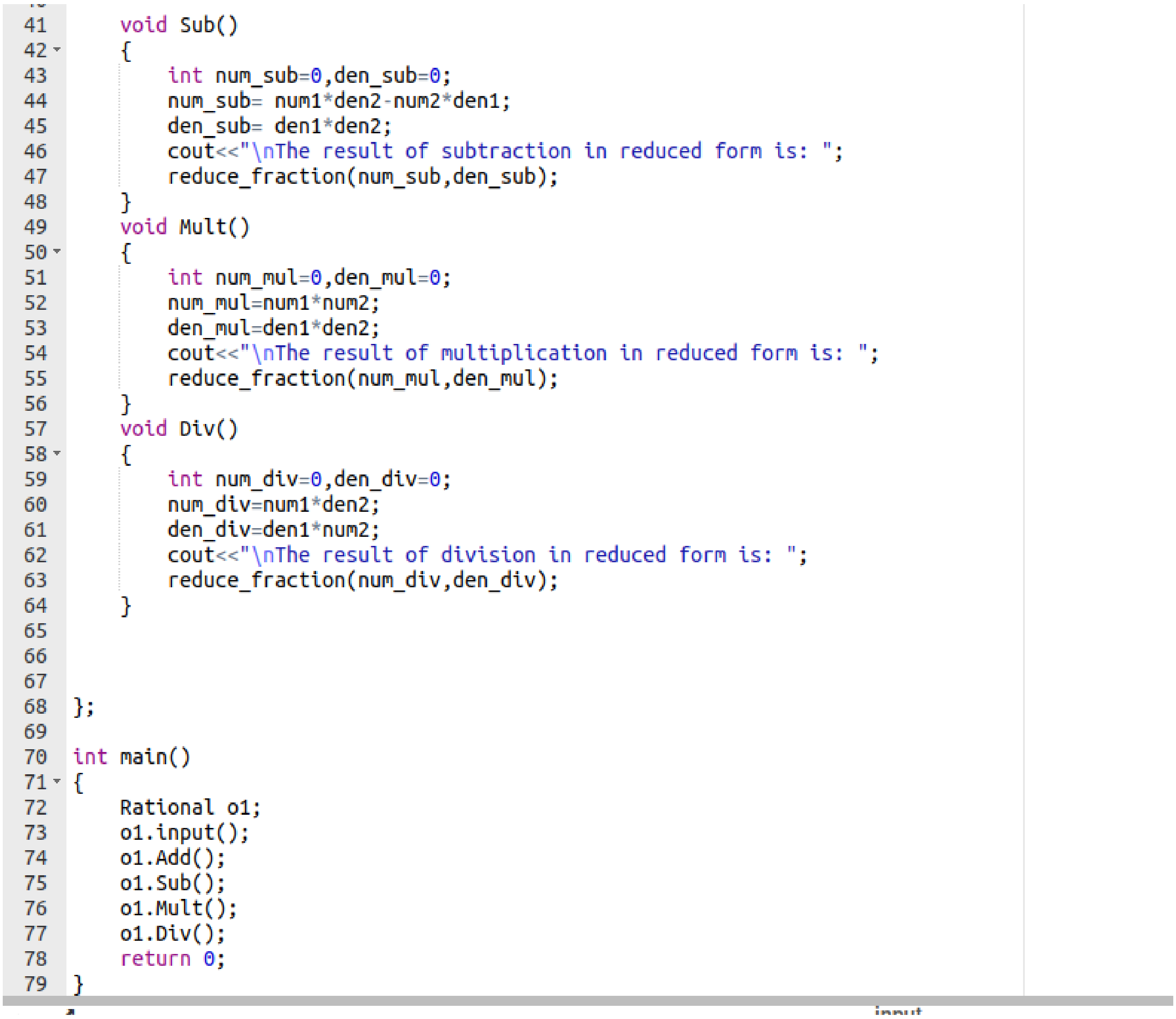
Input



```
#include <iostream>
#include <bits/stdc++.h>
/*Calling this library to call gcd function for highest common factor
 to convert to reduced form*/
using namespace std;

class Rational
{
    int num1,num2,den1,den2;
    public:
    void input()
    {
        cout<<"\n\nEnter the numerator of the first number:";
        cin>>num1;
        cout<<"\nEnter the denominator of the first number:";
        cin>>den1;
        cout<<"\n\nEnter the numerator of the second number:";
        cin>>num2;
        cout<<"\nEnter the denominator of the second number:";
        cin>>den2;
    }
    void reduce_fraction(int a,int b)
    {
      int result=0;
      result = __gcd(a,b);
      /*Calling this gcd function from the above
      library to convert the fraction to simplest form*/
      a=a/result;
      b=b/result;
      cout<<a<<"/"<<b<<endl;
    }
    void Add()
    {
        int num_add=0,den_add=0;
        num_add= num1*den2+num2*den1;
        den_add= den1*den2;
        cout<<"\n\nThe result of addition in reduced form is: ";
        reduce_fraction(num_add,den_add);
    }

    void Sub()
    {
        int num_sub=0,den_sub=0;
        num_sub= num1*den2-num2*den1;
        den_sub= den1*den2;
        cout<<"\nThe result of subtraction in reduced form is: ";
        reduce_fraction(num_sub,den_sub);
    }
```



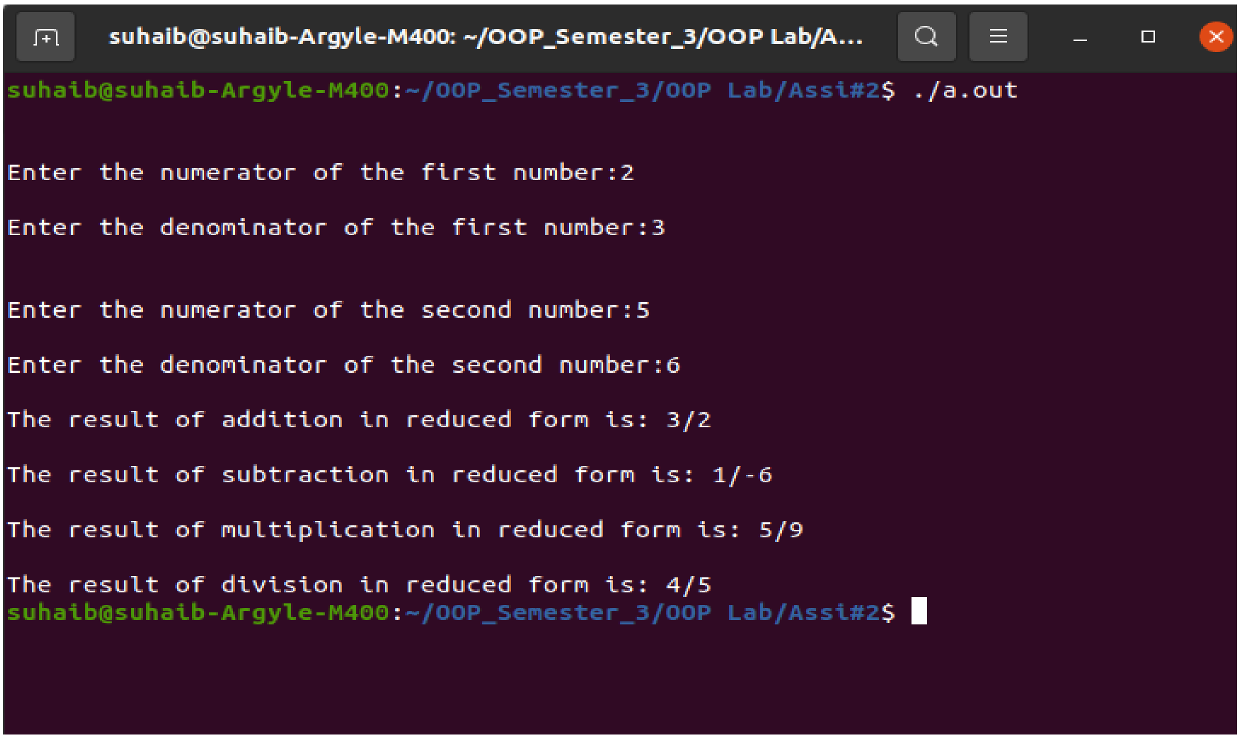
```
    void Sub()
    {
        int num_sub=0,den_sub=0;
        num_sub= num1*den2-num2*den1;
        den_sub= den1*den2;
        cout<<"\nThe result of subtraction in reduced form is: ";
        reduce_fraction(num_sub,den_sub);
    }
    void Mult()
    {
        int num_mul=0,den_mul=0;
        num_mul=num1*num2;
        den_mul=den1*den2;
        cout<<"\nThe result of multiplication in reduced form is: ";
        reduce_fraction(num_mul,den_mul);
    }
    void Div()
    {
        int num_div=0,den_div=0;
        num_div=num1*den2;
        den_div=den1*num2;
        cout<<"\nThe result of division in reduced form is: ";
        reduce_fraction(num_div,den_div);
    }



};

int main()
{
    Rational o1;
    o1.input();
    o1.Add();
    o1.Sub();
    o1.Mult();
    o1.Div();
    return 0;
}
```

Output



```
suhaib@suhaib-Argyle-M400:~/OOP_Semester_3/OOP Lab/Assi#2$ ./a.out


Enter the numerator of the first number:2

Enter the denominator of the first number:3


Enter the numerator of the second number:5

Enter the denominator of the second number:6

The result of addition in reduced form is: 3/2

The result of subtraction in reduced form is: 1/-6

The result of multiplication in reduced form is: 5/9

The result of division in reduced form is: 4/5
suhaib@suhaib-Argyle-M400:~/OOP_Semester_3/OOP Lab/Assi#2$
```

i.Why is it appropriate to set the attributes always private and member functions public?

The main object of object oriented programming to make our code private and restrict others from seeing our code. Thus, we use private attributes. On the other hand, to access the code in our class we use public functions.

ii.Why is it good to make attribute private and access them using setters?

It is good for the privacy of our code.

iii.Can we make member functions private?

Yes we can make the member functions in our code private.

Conclusion:

In this lab I learned how to apply mathematical operations on rational numbers using classes.