**National University of Computer and Emerging Sciences**



**Lab Manual 07**

**Object Oriented Programming**

|  |  |
| --- | --- |
| Course Instructor | Ms. Arooj Khalil |
| Lab Instructor (s) | Ms. Fariha Maqbool  Ms. Amara Nasir |
| Section | BSE-2A |
| Semester | Spring 2023 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## **Objectives**

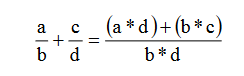
After performing this lab, students will practice:

* Operator Overloading (arithmetic, relational, unary increment (pre/post fix) and stream operators)

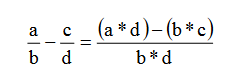
**TASK 1:**

Create a class **RationalNumber** that stores a fraction in its original form (i.e. without finding the equivalent floating pointing result). This class models a fraction by using two data members: an integer for numerator and an integer for denominator. For this class, provide the following functions:

1. A two-argument constructor that initializes the numerator and denominator to the values sent from main. This constructor should prevent a 0 denominator in a fraction, reduce or simplify fractions that are not in reduced form, and avoid negative denominators. If the user skips the values during object creation then it should set the default values of nominator and denominator as 1.
2. An **overloaded** **>>** **operator** that takes the nominator and denominator as input from user. This operator should prevent a 0 denominator in a fraction, reduce or simplify fractions that are not in reduced form, and avoid negative denominators.
3. An **overloaded** **<<** **operator** that displays a fraction in the format a/b.
4. An **overloaded operator +** for addition of two rational numbers. Two fractions a/b and c/d are added together as:

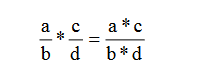


1. An **overloaded operator -** for subtraction of two rational numbers.  
   Two fractions a/b and c/d are subtracted from each other as:



1. An overloaded operator \* for multiplication of two rational numbers.

Two fractions a/b and c/d are multiplied together as:



1. An **overloaded operator ==** that compares the two rational numbers and return true if they are equal and false otherwise
2. An **overloaded prefix operator ++** that increments the values in the nominator and denominator by 1
3. An **overloaded postfix operator –** that decreases the values in the nominator and denominator by 1

int main()

{

RationalNumber RN1;

RationalNumber RN2(5,6);

RationalNumber RN3;

cout<<"Input a rational number"<<endl;

cin>>RN3;

cout<<RN1<<RN2<<RN3;

if(RN2==RN3)

cout<<"RN2 == RN3"<<endl;

else

cout<<"RN2 != RN3"<<endl;

RationalNumber RN4= RN2 - RN3;

cout<<RN4;

RationalNumber RN5= RN2 + RN3;

cout<<RN5;

RN4++;

cout<<RN4;

RN5--;

cout<<RN5;

system("pause");

return 0;

}