Lab 6 **Target: CLO2, CLO3**

Create a class to represent rational numbers and a test program to verify that all required functionality works. In this C++ project you are required to implement a Rational class that represents two integer type rational numbers r1 and r2 to perform arithmetic operations, relational operators, equality operators and unary operators and implemented friend function and operator overloading.

A **RationalNumber** is a type of real number, can be defined as any number which can be represented in the form of p/q where $q \neq 0$. (Where p/0 is not a rational number)

Hint: The code performs several arithmetic operations, binary operators, relational Operators equality operators on r1 and r2 using overloaded operators. Set the limit p/0 is not allowed (condition is false)

Data Members: Two rational number r1 and r2

- **int numerator**
- **int denominator**

Member Functions: numerator and denominator p/q where $q \neq 0$.

- **Constructor:**
    - **Default constructor**
    - **Take two integers: one numerator and one denominator**
- **Accessors:**
    - **int getNumerator()**
    - **int getDenominator()**
- **Mutators:**
    - **void setNumerator(int num)**

- **void setDenominator(int denom)**

- **Operators Overloading**
  - **Binary Operator** (arithmetic, stream operators, relational operators)
    - Using Member functions
    - Using Friend functions

  - **Arithmetic Operators**
    - **+ (Addition):** Adds two **rational numbers** or a rational number and aninteger.
    - **- (Subtraction):** Subtracts two **rational numbers** or a **rational number** and an integer.
    - ***(Multiplication):** Multiplies two **rational numbers** or a **rationalnumber** and an integer.
    - **/ (Division):** Divides a **rational number** by an integer.
    - **% (Modulus):** Calculates the modulus of two **rational numbers** or a **rational number** and an integer.

  - **Comparison Operators**
    - **== (Equal to):** Checks if two **rational numbers** or a **rational number** and aninteger are equal.
    - **!= (Not equal to):** Checks if two **rational numbers** or a **rational number** and aninteger are not equal.
    - **< (Less than):** Checks if a **rational number** is less than another **rational number** or an integer.
      - **(Greater than):** Checks if a **rational number** is greater than another **rational number** or an integer.
      - **<= (Less than or equal to):** Checks if a **rational number** is less than orequal to another **rational number** or an integer.
    - **>= (Greater than or equal to):** Checks if a **rational number** is greater than orequal to another **rational number** or an integer.

  - **Compound Assignment Operators**

- **+=(Addition):** Adds two **rational numbers** or a **rational number** and an integer.
- **-=(Subtraction):** Subtracts two **rational numbers** or a **rational number** and aninteger.
- ***=(Multiplication):** Multiply two **rational numbers** or a **rational number** andan integer.
- **/+(division):** divide two **rational numbers**

- **Unary operator**
  Using member functions
  - **pre-increment operator ++num, post-Increment num++**
  - **pre-decrement operator –num, post-decrement num–**

Your program must be able to handle the following expressions

Code	Expected output
Addition RationalNumber r1(“3/2”); RationalNumber r2(“4/5"); RationalNumber addition=r1+r2; cout<<”Addition:”<<r1+r2;	Addition:23/10
RationalNumber r1(“3/2”); int a=4; cout<<”Addition:”<<r1+a;	Addition:11/2
Subtraction RationalNumbe r1(“3/2”); RationalNumbe r2(“4/5"); RationalNumber subtraction=r1+r2; cout<<”Subtraction:”<<r1-r2;	Subtraction: 7/10
RationalNumbe r1(“3/2”); int a=4; cout<<”Subtraction:”<<r1-a;	Subtraction: -5/2
Multiplication: RationalNumbe r1(“3/2”); RationalNumbe r2(“4/5"); RationalNumber multiplication=r1*r2; cout<<”Multiplication:”<<r1*r2;	Multiplication:6/5
RationalNumbe r1(“3/2”); int a=4; cout<<”Multiplication:”<<r1*a;	Multiplication:6/1 or Multiplication:6

Division RationalNumbe r1(“3/2”); RationalNumbe r2(“4/5"); RationalNumber division=r1/r2; cout<<”Division:”<<r1/r2;	Division:15/8
RationalNumbe r1(“3/2”); int a=4; cout<<”Division:”<<r1/a;	Division:3/8
Comparison operators	
Equal RationalNumber r1(“3/2”); RationalNumbe r2(“4/5"); bool result = (r1 == r2); cout<<”Result:”<<result;	Result:false