```cpp
// Q_1.cpp : This file contains the 'main' function. Program execution begins and ends there.
//
//
//
// /------------------------------------------------------------------------------------------
//Name                          Sai Chaitanya Kilambi
//Course                        CPSC 131 Data Structures, Fall, 2022
//Assignment                    No.14 question:1
//Due date                      12/07/2022
// Purpose:
// This program demonstrates operator overloading
//-------------------------------------------------------------------------------------------
// list of libraries
//
//importing the required libraries

#include <iostream>
using namespace std;

//creating FRACT class
class FRACT {
private:
    int num, den;

public:
    void readFraction()
    {
        cout << "Enter a fraction: ";
        cin >> num;
        cin.ignore();
        cin >> den; // cin.ignore() to skip the “/”
    }
    FRACT friend operator+(FRACT f1, FRACT f2)
    {
        FRACT sum;
        sum.num = (f1.num * f2.den + f1.den * f2.num);
        sum.den = (f1.den * f2.den);
        return sum;
    }
    FRACT friend operator-(FRACT f1, FRACT f2)
    {
        FRACT diff;
        diff.num = (f1.num * f2.den - f1.den * f2.num);
        diff.den = (f1.den * f2.den);
        return diff;
```

```cpp
    }
    FRACT friend operator*(FRACT f1, FRACT f2)
    {
        FRACT prod;
        prod.num = (f1.num * f2.num);
        prod.den = (f1.den * f2.den);
        return prod;
    }
    FRACT friend operator/(FRACT f1, FRACT f2)
    {
        //finding reciprocal of f2 (flipping numerator and denominator)
        FRACT reciprocal;
        reciprocal.num = f2.den;
        reciprocal.den = f2.num;
        //multiplying reciprocal of f2 with f1 to get the result of f1/f2
        return f1 * reciprocal;
    }


    void friend displayFract(FRACT f, bool parentheses = false)
    {
        //if parentheses is true, printing in format (a/b), otherwise
          simply printing
        //in format: a/b
        if (parentheses) {
            cout << "(" << f.num << "/" << f.den << ")";
        }
        else {
            cout << f.num << "/" << f.den;
        }
    }
};

//main function
int main()
{
    FRACT frac1, frac2, fracSum, fracDiff, fracPro, fracDiv;
    frac1.readFraction();
    frac2.readFraction();

    //performing operations
    fracSum = frac2 + frac1;
    fracDiff = frac2 - frac1;
    fracPro = frac2 * frac1;
    fracDiv = frac2 / frac1;

    //display result
    displayFract(frac2);
    cout << " + ";
```

```
 94      displayFract(frac1);
 95      cout << " = ";
 96      displayFract(fracSum);
 97      cout << endl;
 98
 99      displayFract(frac2);
100      cout << " - ";
101      displayFract(frac1);
102      cout << " = ";
103      displayFract(fracDiff);
104      cout << endl;
105
106      displayFract(frac2);
107      cout << " * ";
108      displayFract(frac1);
109      cout << " = ";
110      displayFract(fracPro);
111      cout << endl;
112
113
114      displayFract(frac2, true);
115      cout << " / ";
116      displayFract(frac1, true);
117      cout << " = ";
118      displayFract(fracDiv, true);
119      cout << endl;
120      system("pause");
121      return 0;
122  }
```