```cpp
#include <fstream>
using namespace std;
#include "header.h"

RatNum::RatNum() {
    num = 0;
    den = 1;
}

RatNum::RatNum(int top, int bottom) {
    num = top;
    den = bottom;
    if (bottom == 0){
        num = 0;
        den = 1;
        cout << "Error! Denominator cannot be 0!"; }
}

void RatNum::printvalues(ofstream &outf) {
    outf.setf(ios::fixed);
    outf << "Numerator=" << num << " " << "Denominator=" << den << "    ";
}

int RatNum::getnum(){
    return num;
}

int RatNum::getden(){
    return den;
}

void RatNum::setnum(int top){
    num = top;
}

void RatNum::setden(int bottom){
    den = bottom;
    if (bottom == 0) {
        num = 0;
        den = 1;
        cout << "Error! Denominator cannot be 0!"; }
}

void RatNum::reduceme(){
    if (den < 0) {
        den *= -1;
        num *= -1; }
    if (num == 0) {
        den = 1; }
    if (num == den) {
        num = 1;
        den = 1;}
    if (den == 0) {
        num = 0;
```

```
55            den = 1;
56            cout << "Error! Denominator cannot be 0!";}
57        int stophere = min(abs(num),abs(den));
58        for (int i = stophere; i > 1; i--){
59            if (num % i ==0 && den % i ==0) {
60                num /= i;
61                den /= i;}
62        }
63    }
64
65    RatNum operator -(RatNum x, RatNum y) {
66        RatNum answer;
67        if (x.den != y.den) {
68            int xnum = x.num;
69            int ynum = y.num;
70            int xden = x.den;
71            int yden= y.den;
72            x.num = xnum * yden;
73            y.num = ynum * xden;
74            x.den = xden * yden;
75            y.den = yden * xden; }
76        answer.num = x.num - y.num;
77        answer.den = x.den;
78        answer.reduceme();
79        return answer;
80    }
81
82    RatNum operator +(RatNum x, RatNum y) {
83        RatNum answer;
84        answer.num = x.num * y.den + x.den * y.num ;
85        answer.den = x.den * y.den;
86        answer.reduceme();
87        return answer;
88    }
89
90    RatNum operator /(RatNum x, RatNum y) {
91        RatNum answer;
92        if (y.num == 0) y.den = 1;
93        if (y.den == 0 && y.num == 1) cout << "Error! Cannot divide by zero!";
94        answer.num = x.num * y.den;
95        answer.den = x.den * y.num;
96        answer.reduceme();
97        return answer;
98    }
99
100   RatNum operator *(RatNum x, RatNum y) {
101       RatNum answer;
102       answer.num = x.num * y.num;
103       answer.den = x.den * y.den;
104       answer.reduceme();
105       return answer;
106   }
107
108   istream& operator >>(istream& insert, RatNum& number){
```

```
109         char eat;
110         insert >> number.num >> eat >> number.den >> ws;
111         if (number.den == 0) {
112             number.num = 0;
113             number.den = 1;
114             cout << "Error! Denominator cannot be 0!";
115         }
116         number.reduceme();
117         return insert;
118     }
119
120     ostream& operator <<(ostream& output, RatNum number){
121         if (number.den == 1) {
122             output << number.num; }
123         else output << number.num << "/" << number.den;
124         number.reduceme();
125         return output;
126     }
127
128     int min(int a, int b){
129         if (a < b) return a;
130         else return b;
131     }
132
```