

Source File: ~/2305/lab02.(C|CPP|cpp|c++|cc|cxx|cp)
Input: Under control of main function
Output: Under control of main function
Value: 2

Create a class called `Rational` for performing various operations with fractions. The specification of the class will be provided. Your task will be to provide the implementation. A main program for testing your implementation will also be provided.

Use a two-element array of `ints` to represent the `private` data of the class—the numerator and the denominator. If the array is declared as

```
int r[2];
```

use `r[0]` to represent the numerator and `r[1]` the denominator. The implementation should provide two constructors: a default constructor that initializes the numerator to zero and the denominator to one and a second constructor that takes two arguments (the first argument should be stored in the numerator and the second in the denominator). The constructors should **not** store the rational number in reduced form. Additional `public` member functions include:

- “Set” functions for setting the numerator and denominator. The `setDenominator` function should check its argument for validity. If the function receives an argument equal to zero (0), the function should set the denominator to one (1).
- “Get” functions for getting the numerator and denominator.
- Reduction of a `Rational` to lowest terms. Also, `reduce` should modify the denominator of a `Rational` with a zero numerator to be one. Further, a negative `Rational` should ensure that the numerator is negative and the denominator is positive. `Rationals` having both numerator and denominator negative should be modified such that both numerator and denominator are positive.

If u and v are integers, not both zero, we say that their greatest common divisor, $\text{gcd}(u, v)$, is the largest positive integer that evenly divides both u and v . When u and v are both zero, every integer evenly divides zero, so it is convenient to set $\text{gcd}(0, 0) = 0$. When either u or v is zero, define $\text{gcd}(u, 0) = |u|$ and $\text{gcd}(0, v) = |v|$. Provide the implementation of this function as a `private` member of the `Rational` class.

A header file is shown in Figure 1, a sample `main` function for testing your implementation is shown in Figure 2, a sample data file is shown in Figure 3, and a sample execution sequence is shown in Figure 4. To use the `Makefile` as distributed in class, add a target of `lab02` to `targets2srcfiles`.

```

1  #ifndef LAB02_H
2  #define LAB02_H
3
4  using namespace std;
5
6  class Rational
7  {
8  public:
9      Rational();                // default constructor
10     Rational(int num, int denom); // additional constructor
11     void setNumerator(int num);   // set numerator to num
12     void setDenominator(int denom); // set denominator to denom
13     int getNumerator() const;     // return numerator
14     int getDenominator() const;   // return denominator
15     void reduce();                // Reduce to lowest terms and
16                                 // normalize

```

Figure 1. `/usr/local/2305/include/lab02.h` (Part 1 of 2)

```
17 private:
18     int r[2];                // r[0] represents numerator
19                               // r[1] represents denominator
20     int gcd(int m, int n) const; // returns the gcd of m and n
21 };
22
23 #endif
```

Figure 1. /usr/local/2305/include/lab02.h (Part 2 of 2)

```
1 #include <lab02.h>
2 #include <iostream>
3 #include <cstdlib>
4
5 using namespace std;
6
7 int main()
8 {
9     int n, d;
10    Rational first(1, -2), second(-3, 0), third;
11
12    cout << "first = " << first.getNumerator() << '/' << first.getDenominator()
13         << endl;
14    cout << "second = " << second.getNumerator() << '/'
15         << second.getDenominator() << endl;
16    cout << "third = " << third.getNumerator() << '/' << third.getDenominator()
17         << endl;
18
19    while (cin >> n >> d)
20    {
21        third.setNumerator(n);
22        third.setDenominator(d);
23        cout << "Before Reduce() third = "
24             << third.getNumerator() << '/' << third.getDenominator();
25        third.reduce();
26        cout << " After Reduce() third = "
27             << third.getNumerator() << '/' << third.getDenominator() << endl;
28    }
29
30    return EXIT_SUCCESS;
31 }
```

Figure 2. /usr/local/2305/src/lab02main.C

```

1  -3  4  3  4
2   3 -4 -3 -4
3  25 45  8 99
4   1  0  2  0
5  129 6579 1935 249
6  1331 1651 2301 1079
7   3 1260 6 198
8  43 1935 207 6579
9   5 7 -25 -35
10 -83 1651 127 -1079
11 1079 1651

```

Figure 3. /usr/local/2305/data/02/01.dat

```

1 2305-> cp /usr/local/2305/src/lab02main.C .
2 2305-> make lab02
3 g++ -g -Wall -c lab02main.C -I/usr/local/2305/include -I.
4 g++ -g -Wall -c lab02.C -I/usr/local/2305/include -I.
5 g++ -o lab02 lab02main.o lab02.o -L/usr/local/2305/lib -lm -lbits
6 2305-> cat /usr/local/2305/data/02/01.dat | lab02
7 first = 1/-2
8 second = -3/1
9 third = 0/1
10 Before Reduce() third = -3/4 After Reduce() third = -3/4
11 Before Reduce() third = 3/4 After Reduce() third = 3/4
12 Before Reduce() third = 3/-4 After Reduce() third = -3/4
13 Before Reduce() third = -3/-4 After Reduce() third = 3/4
14 Before Reduce() third = 25/45 After Reduce() third = 5/9
15 Before Reduce() third = 8/99 After Reduce() third = 8/99
16 Before Reduce() third = 1/1 After Reduce() third = 1/1
17 Before Reduce() third = 2/1 After Reduce() third = 2/1
18 Before Reduce() third = 129/6579 After Reduce() third = 1/51
19 Before Reduce() third = 1935/249 After Reduce() third = 645/83
20 Before Reduce() third = 1331/1651 After Reduce() third = 1331/1651
21 Before Reduce() third = 2301/1079 After Reduce() third = 177/83
22 Before Reduce() third = 3/1260 After Reduce() third = 1/420
23 Before Reduce() third = 6/198 After Reduce() third = 1/33
24 Before Reduce() third = 43/1935 After Reduce() third = 1/45
25 Before Reduce() third = 207/6579 After Reduce() third = 23/731
26 Before Reduce() third = 5/7 After Reduce() third = 5/7
27 Before Reduce() third = -25/-35 After Reduce() third = 5/7
28 Before Reduce() third = -83/1651 After Reduce() third = -83/1651
29 Before Reduce() third = 127/-1079 After Reduce() third = -127/1079
30 Before Reduce() third = 1079/1651 After Reduce() third = 83/127
31 2305->

```

Figure 4. Commands to Compile, Link, & Run Lab 02