CS 1400-03 Introduction to Programming and Problem Solving
Project #5
(Due: 11:59 PM, Monday, 4/5/2021)

Write a class `Fraction` for representing fractions and performing arithmetic with them. The `Fraction` class has the following private fields:

`numerator`: an integer
`denominator`: an integer
`created`: a `static` integer to count the number of fractions that are created so far

The class should have the following public methods:

`constructor`: A no-argument constructor that sets the default fraction to 0/1
`constructor`: A 2-argument constructor that accepts a numerator and a denominator as arguments
`add`: performs addition of two fractions and returns a new fraction
`subtract`: performs subtraction of two fractions and returns a new fraction
`multiply`: performs multiplication of two fractions and returns a new fraction
`divide`: performs division of two fractions and returns a new fraction
`toString`: returns a string for the fraction in a form like 2/3
`printAsFloat`: prints the fraction in a form like 0.66667
`numberOfFractions`: A `static` method to return the number of fractions which have been created so far
`read`: a `static` method to prompt the user to enter the numerator and denominator of a fraction; the method returns the created fraction.

Your constructor should take care of the following cases:

- Prevent creation of fractions with 0 in the denominator. Print a warning message and change the fraction to 0/1.
- Anything with 0 in the numerator and non-zero denominator — *e.g.* 0/-1 or 0/2 — should reduce to 0/1.
- Negative signs, if any, should appear only in the numerator. Thus, -1/2 is acceptable, but 1/-2 is not and should be changed to -1/2.
- The numerator and denominator must not have any common divisors. The following code computes the greatest common divisor (gcd) of two integers. In order to reduce a fraction to lowest terms, you should divide both the numerator and denominator by their gcd.

```
// computes and returns the gcd of the two positive parameters
// by the Euclid's algorithm
private int gcd (int num1, int num2)
{
   while (num1 != num2)
   {
      if (num1 > num2)
         num1 = num1 - num2;
      else
         num2 = num2 - num1;
   }
   return num1;
}
```

Write a separate `private` method to normalize data as described above, and have this method be called by the 2-argument constructor.

Example. Driver code like this —

```
Fraction a = new Fraction(1,3);
Fraction b = new Fraction(2,4);
Fraction c = new Fraction();
System.out.println("c = " + c);

c = a.add(b);
System.out.print(a + " + " + b + " = " + c + " = ");
c.printAsFloat();
System.out.println();
...
```

would produce output like this —

```
c = 0/1
1/3 + 1/2 = 5/6 = 0.8333333
...
```

Write a driver program, called `FractionDemo.java`, that will ask the user to enter two fractions, perform addition, subtraction, multiplication, and division of the two fractions, and display the results. The program will ask the user if he or she wishes to perform the operation again. If the answer is 'Y' or 'y', these steps should repeat. Otherwise, the program will display the total number of fractions that have been created so far and terminate. The following are **required** I/O behavior when running the program, where the user's input is shown in bold:

```
fcsang@abbott ~/cs1400/2020fall/project $ java FractionDemo
Please enter 2 fractions --
Fraction 1:
enter an integer numerator: 1
enter an integer denominator: 2
Fraction 2:
enter an integer numerator: 1
enter an integer denominator: 3
1/2 + 1/3 = 5/6 = 0.8333333
1/2 - 1/3 = 1/6 = 0.16666667
1/2 * 1/3 = 1/6 = 0.16666667
1/2 / 1/3 = 3/2 = 1.5

Do you want to continue ('Y' or 'y' for yes)? n
6 fractions have been created.

fcsang@abbott ~/cs1400/2020fall/project $ java FractionDemo
Please enter 2 fractions --
Fraction 1:
enter an integer numerator: 3
enter an integer denominator: 0
denominator cannot be 0
the fraction is set to 0/1
Fraction 2:
enter an integer numerator: 6
enter an integer denominator: -2
0/1 + -3/1 = -3/1 = -3.0
0/1 - -3/1 = 3/1 = 3.0
0/1 * -3/1 = 0/1 = 0.0
```

```
0/1 / -3/1 = 0/1 = 0.0

Do you want to continue ('Y' or 'y' for yes)? y
Please enter 2 fractions --
Fraction 1:
enter an integer numerator: 0
enter an integer denominator: -5
Fraction 2:
enter an integer numerator: -5
enter an integer denominator: -20
0/1 + 1/4 = 1/4 = 0.25
0/1 - 1/4 = -1/4 = -0.25
0/1 * 1/4 = 0/1 = 0.0
0/1 / 1/4 = 0/1 = 0.0

Do you want to continue ('Y' or 'y' for yes)? n
12 fractions have been created.
```

In your `cs1400/project` directory, create two programs named `Fraction.java` and `FractionDemo.java`. Your Java programs must begin with the comments below and follow the naming and coding conventions posted on Blackboard.

```
// your name
// CS1400, section 03
// Project 5 – Arithmetic Operations with Fractions
// date
```

Generate a `script` file `pj5.txt` with appropriate time stamps and the following steps visible:

1. a `pwd` to show the current working directory
2. an `ls -l` to show in long format the files in your `cs1400/project` directory
3. display both `Fraction.java` and `FractionDemo.java`
4. compile `FractionDemo.java`
5. run `FractionDemo` **eight times with the following test cases**: (1/2, 1/3), (3/0, 6/-2), (0/-5, -5/-20), (12/36, 70/84), (-8/2, 10/10), (13/-1, 0/9), (1212/123, -123/-1212), (3/-20, -2/9)

Submit the file `pj5.txt` on Gradescope.