

Programming Assignment 10-2

A *rational number* is a real number that can be represented as a fraction of the form p/q , where $q > 0$ and both p and q are integers. Although Java has data types for integers and floating point numbers, it does not have an explicit data type for rational numbers. Create a Java class `Rational` that meets this need. Design so that it meets the following requirements:

- The constructor accepts two integers p, q ; if q is not positive, an error is indicated with a statement printed to the console
- The class has the following methods, with these signatures:

```
//adds the rational rat to this Rational
public Rational add(Rational rat);
```

```
//multiplies rat by this Rational
public Rational multiply(Rational rat);
```

```
//returns -1 if this rational is less than rat
//returns 0 if this rational equals (see equals
//      method discussion below) rat
//returns 1 if this rational is greater than rat
public int compareTo(Rational rat)
```

Mutators and accessors for numerator and denominator

- It overrides the `equals()` method. Recall that two rationals p/q and r/s are equal if and only if $qr = ps$. (Therefore, you must override `hashCode()` too.)
- It overrides the `toString()` method. The return value of the `toString()` method should represent the Rational in the usual format; for instance, the rational having numerator 2 and denominator 3 should look like this:
 $2/3$
- In the main method of your class, test your methods by performing the following computations:

$$(2/3 * -17/5) + 1/3,$$
$$2/3 * (-17/5 + 1/3),$$

and then stating which of the two values is larger. All computations and output values should involve *fractions only* – no floating point numbers allowed.

Expected output:

`(2/3 * -17/5) + 1/3 is greater than 2/3 * (-17/5 + 1/3)`

Hint:

$a/b < c/d$ if and only if $ad < bc$