Classes and Objects 1

Consider the Fraction that we have created in class.

1. Complete the definitions of the following instance methods for the Fraction class.

   a) `public Fraction times (Fraction other)`
   This method returns a `Fraction` object that is the product of the implicit object parameter and the explicit parameter `other`.

   b) `public Fraction plus (Fraction other)`
   This method returns a `Fraction` object whose value is the sum of the implicit object parameter and the explicit parameter `other`. The method should leave both its explicit and implicit parameters unchanged.

2. Assume the `larger` method is declared as followed:

```
public Fraction larger (Fraction other) {
   if (this.size() >= other.size() ) {
      return this;
   } else {
      return other;
   }
}
```

   Suppose that p, q, and r are all objects of type Fraction. What fraction would r represent after the statement

```
r = p.larger(q);
```

   is executed?

   a) `p` represents $\frac{1}{3}$ and `q` represents $\frac{4}{5}$
   b) `p` represents $\frac{7}{-5}$ and `q` represents $\frac{-9}{-7}$
   c) `p` represents $\frac{5}{6}$ and `q` represents $\frac{-25}{-30}$
   d) `p` represents $\frac{-9}{-12}$ and `q` represents $\frac{-3}{4}$
   e) `p` represents $\frac{5}{8}$ and `q` represents $\frac{13}{20}$

3. Complete the definitions of the following instance methods for the Fraction class.

a) `public void timesEquals(Fraction p)`
This method should have the same effect (for `Fraction` objects) that the *= operator has for primitive numeric types. Thus, if called by the statement

```
p.timesEquals(q);
```

(where `p` and `q` are objects of type `Fraction`), the method would make `p` represent the product of the fractions currently represented by `p` and `q` while the value of `q` would be left unchanged.

b) `public Fraction plus (Fraction f)`
The method should return a `Fraction` object whose value is the sum of the implicit object parameter (`this`) and the explicit parameter (`f`). The method should leave both its explicit and implicit parameters unchanged.

c) `public void reduce()`
The method should reduce its implicit Fraction parameter to lowest terms. For example, if f represents the fraction $\frac{16}{24}$, the statement

```
f.reduce();
```

should change `f` so that it represents the fraction $\frac{2}{3}$.