

## Lab 3: Creating an Abstract Data Type

In this assignment you will write your own class definition that implements an abstract data type (ADT) for rational numbers.

A rational number is any value  $p/q$ , where  $p$  and  $q$  are both integers and  $q$  is not zero.  $p$  is called the *numerator* and  $q$  is called the *denominator*. Objects of your class should be immutable once they have been instantiated.

Your class is to be called `Rational` and must include the following public methods:

Method
<code>Rational(int numerator, int denominator)</code> Constructor: if numerator and denominator are not in lowest terms, this method should use Euclid's algorithm to reduce them prior to storing them in instance variables Requires: denominator $\neq 0$
<code>Rational plus(Rational b)</code> Ensures: return value is the sum of this and b
<code>Rational minus(Rational b)</code> Ensures: return value is the difference of this and b
<code>Rational times(Rational b)</code> Ensures: return value is the product of this and b
<code>Rational divides(Rational b)</code> Ensures: return value is the ratio of this and b
<code>boolean equals(Rational b)</code> Ensures: return value is this = b
<code>long numerator()</code> Ensures: return value is the numerator of this
<code>long denominator()</code> Ensures: return value is the denominator of this
<code>String toString()</code> Ensures: return value is a string representation of $n/d$ , where $n$ is the numerator and $d$ is the denominator

You should include a main method in your class that is a test client that demonstrates each method working properly. Also include Javadocs comments in your program that contain the contract specifications.

Remember that individual work is expected on the lab projects!

**To Do**

1. Determine the instance variables required.
2. Stub out the public methods detailed in the specifications.
3. Implement each method according to the specifications.
4. Comment the class and each method with javadoc style comments.

<b>Points</b>	<b>Item</b>	<b>Description</b>
5	documentation	Use of comments including Javadoc-style comments, use of self-documenting identifier names, and judicious use of white space (such as indentation) to enhance readability.
3	representation	The instance variable(s) that you declare to represent rational number.
12	results	The constructor, plus, and minus are each worth 2 points. All other methods are worth 1 point each.

**To Submit:** Email your Rational.java source file as an attachment to [dstucki@otterbein.edu](mailto:dstucki@otterbein.edu)