

## Assignment 3

A rational number is a number that can be represented as the ratio of two integers. For example,  $2/3$  is a rational number, and you can think of 7 as a rational number with an implicit 1 in the denominator. For this assignment, you are going to write a class definition for rational numbers.

1. Create a new program called `Rational.java` that defines a class named `Rational`. A `Rational` object should have two integer instance variables to store the numerator and denominator.
2. Write a constructor that takes no arguments and that sets the numerator to 0 and denominator to 1.
3. Write a method called `printRational` that takes a `Rational` object as an argument and prints it in some reasonable format.
4. Write a `main` method that creates a new object with type `Rational`, sets its instance variables to some values, and prints the object.
5. At this stage, you have a minimal testable program. Test it and, if necessary, debug it.
6. Write a second constructor for your class that takes two arguments and that uses them to initialize the instance variables.
7. Write a method called `negate` that reverses the sign of a rational number. This method should be a modifier, so it should return `void`. Add lines to `main` to test the new method.
8. Write a method called `invert` that inverts the number by swapping the numerator and denominator. Add lines to `main` to test the new method.
9. Write a method called `toDouble` that converts the rational number to a double (floating-point number) and returns the result. This method is a pure function; it does not modify the object. As always, test the new method.
10. Write a modifier named `reduce` that reduces a rational number to its lowest terms by finding the greatest common divisor (GCD) of the numerator and denominator and dividing through. This method should be a pure function; it should not modify the instance variables of the object on which it is invoked. To find the GCD, see Exercise 10).
11. Write a method called `add` that takes two `Rational` numbers as arguments and returns a new `Rational` object. The return object should contain the sum of the arguments.

There are several ways to add fractions. You can use any one you want, but you should make sure that the result of the operation is reduced so that the numerator and denominator have no common divisor (other than 1).

The purpose of this exercise is to write a class definition that includes a variety of methods, including constructors, modifiers and pure functions.