

# CSCI 310, Data Structures, Summer 2009

## Assignment 5 (15 points)

1. (15 points) Create a class called `Rational` for performing arithmetic with fractions. Write a tester program for your class.

Use integer variables to represent the private data of the class – the numerator and the denominator stored in simplified form. Your class must perform data validation to ensure the denominator is not zero. At the minimum, your class should include the following (you may need to provide additional helper functions):

### Constructors

1. Default constructor that sets the `Rational` to 0/1.
2. A copy constructor.
3. A constructor that takes two integer values, one for the numerator and the other for the denominator.
4. A constructor that takes just one integer, which is the numerator. The denominator is set to 1.

### Mutator Functions

1. Set the numerator.
2. Set the denominator.
3. Set the numerator and denominator.

### Accessor Functions

1. Get the numerator.
2. Get the denominator.
3. Get a double value that is a decimal representation of the `Rational` number.
4. Get a new object of the `Rational` class that is the reciprocal.

### Predicate Functions

1. Is the `Rational` number equal to zero.
2. Is the `Rational` number positive.
3. Is the `Rational` number negative.
4. Pass two integers (numerator and denominator) to a `Rational` number and see if the `Rational` number has those values as its numerator and denominator.

## Overloaded Operators

The `Rational` should overload the following operators.

1. Stream insertion and stream extraction operators `<<` and `>>`.
2. Relational operators `<`, `<=`, `==`, `>=`, `>`, `!=`.
3. Assignment operator `=`.
4. Compound assignment operators `+=`, `-=`, `*=`, `/=`.
5. Binary arithmetic operators `+`, `-`, `*`, `/`.
6. Unary pre/post increment/decrement operators `++rat`, `--rat`, `rat++`, `rat--`.
7. Unary plus and minus `+`, `-`.