## Assignment 5

Please complete the problems listed below. This assignment forms part of the assessment for this module. You are required to upload a copy of your work on Moodle. Please upload only a single **java** file. You must include as header your name, student number and the assignment number as header for this java file.

A rational number is any number that can be expressed as the quotient or fraction of two integer values, with the denominator not equal to 0. That is, any number that can be written in the form . Everyone learned to work with fractions at some stage in their youth so we just present, without comment, a list of operators and their meaning.

We will use these definitions when we implement operators for our rational numbers.

An interesting observation about rational numbers is that equal values can have different forms. For example, The normalized value of all of these fractions is because each of the other values can be simplified to this value.

Another observation, from a programming perspective, is that rational numbers are immutable. Each number is fixed and there is no operation that magically transforms it into some other value. Similarly, you can add one rational number to another one the result of the addition is a new rational number. The original values used in the addition are not changed or modified.

The class Rational should be declared final because we do not want to allow sub-classes to be created. It must implement the Operations interface listed below.

interface Operations{

public Rational add(Rational q);

public Rational mult(Rational q);

public Rational multBy(int k);

public Rational divBy(Rational q);

public boolean le(Rational q); //returns true if this less than q

public boolean eq(Rational q); //returns true if this equals q

public boolean gt(Rational q); //returns true if this greater than q

public Rational min(Rational q); //returns min of this and q

public Rational max(Rational q); //returns max of this and q

}

The class, as presented below, has a single constructor that takes two integer arguments It assumes that the denominator, d, is not zero. It also has two attributes, num and den, that refer to the *numerator* and *denominator* of the fraction. The private function gcd, greatest common divisor, is used to ensure that all fractions are stored in normalized form. This function calculates the greatest common divisor using only the absolute values of both the numerator and denominator. Both n and d are divisible by g.

class Rational implements Operations{

private int num,den;

public Rational(int n, int d){//assume d != 0

if(n < 0 && d < 0){ n = -n; d = -d;}

else if(d < 0){ n = -n; d = -d;}

//ensures that d never negative and n positive e.g. 1/-2 is changed to -1/2

int g = gcd(Math.abs(n), Math.abs(d));

num = n/g;den = d/g;

}

public Rational(int n){//d == 1 ...}

public int num(){return num;}

public int den(){return den;}

public Rational add(Rational q){...}

public Rational mult(Rational q){...}

// ... all the other required methods

public String toString(){...}

private int gcd(int a, int b){

if(b == 0) return a;

else return gcd(b,a%b);

}

}