

# CS0045 ALGORITHMS AND INFO STRUCTURES APPLICATIONS

## Assignment 1

### Task

Create a class, **Rational**, to represent a fraction – an integer over an integer. The fraction may not necessarily be reduced. You must follow the UML below:

Rational
- numerator: int - denominator: int
+ Rational() // sets numerator to 0, denominator to 1 + Rational(i: int) // sets numerator to i, denominator to 1 + Rational(n: int, d: int) + getNumerator(): int + getDenominator() : int + setNumerator(n: int): void + setDenominator(d: int): void + invert(): void // switch values of numerator and denominator + toDecimal(): double // returns the value of the Rational as a decimal + toString(): String // returns the Rational as a String // in format like 2/3

When Rational is complete and compiles correctly, download the file **TestRational.java** and compile it. Run **TestRational.class** to test whether you have created the **Rational** class correctly. **Do not change the TestRational file.**

### Turning in the Assignment

When you have finished upload the files **Rational.java**, **Rational.class** and a screenshot of **TestRational** executing. Due by the start of the next class.

### Bonus

Write a **void** method **reduce** that will reduce the **Rational** to its lowest terms. Show that it works by adding the following four lines to the end of **TestRational** and uploading a copy of the output:

```
r1.reduce();
r2.reduce();
r3.reduce();
System.out.println(r1 + " " + r2 + " " + r3);
```