

Rational Class with Exceptions and Testing

This lab exercise contains starter files with a skeleton for a class called **Rational**. However, you should copy and paste the contents of the methods from last week's lab methods into that file (don't copy and paste the file, copy and paste the contents of each method one by one). Your tasks this week will be to practice JUnit tests, make the Rational class officially immutable, add a few new methods and constructors, and add some exception throwing.

Stages:

- (1) We have given you a file **StudentTests.java** in which you should implement the tests within.
- (2) Add the keyword **final** in front of the private numerator and denominator fields and see if anything else needs to change as a result.
- (3) Implement a **single-parameter constructor** and a **copy constructor**.
- (4) Implement the new instance **multiply** method and the **subtract** method.
- (5) Consider each method of the class and decide whether it needs to have a test for a zero as the denominator of some **Rational** object that it will create. If it does, add that test and the line of code: **throw new ArithmeticException("Divide by Zero");**
- (6) Write some new test cases to see if this exception is thrown when it should be. For example, the following will try to construct an invalid **Rational** object. If it succeeds in creating that illegal object, the test fails due to the **assertTrue(false);** statement. If the attempt throw the exception, the test will pass due to the **assertTrue(true);** statement.

```
@Test
public void testConstructorOnDivideByZero() {

    try {
        Rational rationalNum = new Rational(3, 0);
        assertTrue(false);
    }
    catch (ArithmeticException e) {
        assertTrue(true); //Not needed, but adds to readability.
    }

}
```