

Homework 2

EE 363 (Fall 2019)

Department of Electrical and Computer Engineering
Clarkson University

Instructions

Please read the instructions carefully before submitting your work.

Note: There is 1 question in this homework for a total of **50** points.

Note: Solve all problems and upload your answers to Moodle using the instructions below. Whenever you write solutions on paper, you need to scan all documents before submitting.

Note: **USR** stands for your login ID on Polaris (`polaris.clarkson.edu`). Make sure any code you write works on Polaris before uploading your files. You will likely lose many points if your code doesn't compile.

Note: **MJR** represents your major, i.e. CE, SE, EE, CS, etc.

Note: Do not upload any executable or intermediate files as answers to problems, unless specifically asked to do so.

Note: You should upload a single ZIP file, named **USR_HW2_MJR.zip**, to Moodle. Inside this, there should be a directory named **hw2**, which in turn should have a directory **p1** with the appropriate deliverable.

1. [50 points] Read §1.2 of the textbook. Solve problem 1.2.17 (immutable `Rational` type with overflow validation). The class should be in package `hw2.p1`. A sample interaction with the program is shown below. (Note that the lines enclosed in square brackets shows the part of the source code required for understanding the output.) Make sure to test that your program can be compiled & executed on Polaris in the same way as shown below.

```
$ ls -R hw2/
hw2/:
p1

hw2/p1:
Rational.java
$ javac hw2/p1/Rational.java
$ java -ea hw2.p1.Rational
r1: 5/6
r2: 3/4
r1 + r2: 19/12
r1 - r2: 1/12
r1 * r2: 5/8
r1 / r2: 10/9

[Rational cr1 = r1;]
[Rational ccr1 = new Rational(r1.numerator(), r1.denominator());]
cr1: 5/6
ccr1: 5/6
n5: 7/8
r1 equals cr1: true
r1 equals ccr1: true
r1 equals equals n5: false

Overflow test:
Integer.MAX_VALUE: 2147483647
r3: 2147483647/3
r4: 10/1
Will attempt r3*r4 ...
Exception in thread "main" java.lang.AssertionError: Overflow detected
    at hw2.p1.Rational.times(Rational.java:91)
    at hw2.p1.Rational.main(Rational.java:173)
```

When assertions are disabled, the program should cause an overflow as shown below:

```
Overflow test:
Integer.MAX_VALUE: 2147483647
r3: 2147483647/3
r4: 10/1
Will attempt r3*r4 ...
r3*r4 = -10/3
```

Deliverable: `Rational.java`.