1. **(10 points) Chapter 8: Self-check problems #7, #11, #15, #19, #26, #27.**

For your benefit, I suggest you complete all of the self-check problems on your own. However, I will just collect the answers to the above six problems. Please type the answers into a document.

a. NO NEED to copy original questions, just put down the question number and the answer.
b. Name the file as "***LastnameFirstname8.doc***" (or ".docx" or ".pdf", where "Lastname" is your last name, and "Firstname" is your first name), submit it online.

2. **(30 points) Chapter 8: Programming Project #1 "Rational Number"**

a. Download the attached "RationalNumber.java"

b. Background: a rational number represents a fraction, e.g. 3 / 5. Here, 3 is the numerator, and 5 is the denominator; both are integers (can be either positive or negative). The denominator cannot be 0.

c. Implement two constructors:

```
public RationalNumber(int numerator, int denominator)
```

It constructs a new rational number. You can assume that the denominator will never be 0.

```
public RationalNumber()
```

It constructs a new rational number to represent 0 / 1

d. Implement the two accessors:

```
public int getDenominator()
```

It returns this rational number's denominator. For example, for 3 / 5, it returns 5.

```
public int getNumerator()
```

It returns this rational number's numerator. For example, for 3 / 5, it returns 3.

e. Implement the `toString` method:

```
public String toString()
```

-- It returns a String representation of this rational number, such as "3/5".
-- If the denominator is 1, omit it. For example, for 4 / 1, it returns "4".
-- If the numerator is 0, omit denominator. For example, for 0 / 8, it returns "0".
-- If both of the numerator and the denominator are negative, return positive. For example, for -3 / -5, it returns "3/5".

-- If the numerator is positive and the denominator is negative, return the negative sign in front of the numerator. For example, for 3 / -5, it returns “-3/5”
-- You do NOT need to worry about reduced form (e.g. for 3 / 6, just returns “3/6” is good enough, no need to return “1/2”).

f. Implement the methods to add, subtract, multiply and divide another rational number. Here is the “method signature” for add:
   `public RationalNumber add(RationalNumber numberToAdd)`

   In the method, you should compute and return the result for the addition. For example, if r1 is 3 / 5, r2 is -1 / 5, **`r1.add(r2)`** should return a value of 2 / 5.

   Whether you choose to update the value of r1 is your choice -- in other words, it won’t affect your grade, as long as the value returned from the method is correct.

   Again, you don’t need to worry about reduced form.

   Please do NOT change any “method signatures” in the original file, otherwise you will receive significant penalty.

g. For this assignment, **you are limited to the language features in Chapters 1 through 8 of the textbook.**

h. Remember, your program will be graded both on “external correctness” and “internal design and style” (whether your source code follows the **style guide**)

i. Submit the final “**`RationalNumber.java`**” file **(DO NOT change the file name)**