

AP Computer Science	Lab 4.1 Rational Objects
The Rational Class Program I	70, 80, 90 & 100 Point Versions
Assignment Purpose: The primary purpose of this lab is to demonstrate knowledge of creating a class with object methods, instantiate multiple objects of the created class, and then call the object methods from the main program method.	

Write a program with a **Rational** class. The purpose of the **Rational** class is to manipulate rational number operations. A rational number is a number that can be expressed in the form **A / B** where **A** and **B** are both whole numbers (no fractions or decimals) and **B ≠ 0**.

The **main** method is provided for you and needs to be used as shown. You are also provided with a **getGCF** method of the **Rational** class which will return the Greatest Common Factor of 2 integers. You will find this useful in writing other methods of the **Rational** class. Your mission is to complete the **Rational** class that is used by this program.

70 Point Version Specifics

Your **Rational** class needs to declare two data attributes: **originalNumerator** for the numerator and **originalDenominator** for the denominator. Only one constructor is required, which uses two parameters entered in the console. The first parameter is the numerator and the second parameter is the denominator. The 70 point version requires you to complete a few additional methods. You will need to complete the **getNum**, **getDen**, **getOriginal**, and **getDecimal** methods.

- **getOriginalNumerator()** returns the **originalNumerator**.
- **getOriginalDenominator()** returns the **originalDenominator**.
- **getOriginalRational()** returns a string representation of the original fraction, like "3/4".
- **getDecimal()** returns the decimal equivalent of the fraction.

70 Point Version Outputs

```

Problems @ Javadoc Declaration
<terminated> RationalTestPart1 [Java Applica

Enter the numerator ----> 3

Enter the denominator --> 4
|
3/4 equals 0.75

```

```

Problems @ Javadoc Declaration
<terminated> RationalTestPart1 [Java Applic

Enter the numerator ----> 2

Enter the denominator --> 9

2/9 equals 0.2222222222222222

```

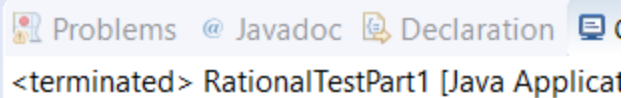
80 Point Version Specifics

The 80 point method adds 2 new attributes to the object: `reducedNumerator` and `reducedDenominator`. These attributes are given value through a private helper method that you will create called `reduce()`. The last method to add is the `getRational()` method, which returns a string representation of the reduced fraction.

You need to change your display data method to read as it does below.

```
public void displayData()
{
    System.out.println();
    System.out.println(getOriginal() + " equals " + getDecimal());
    System.out.println();
    System.out.println("and reduces to " + getRational());
    System.out.println();
}
```

80 Point Version Output



 Problems @ Javadoc  Declaration  Search

<terminated> RationalTestPart1 [Java Applicat]

Enter the numerator ----> 6

Enter the denominator --> 8

6/8 equals 0.75

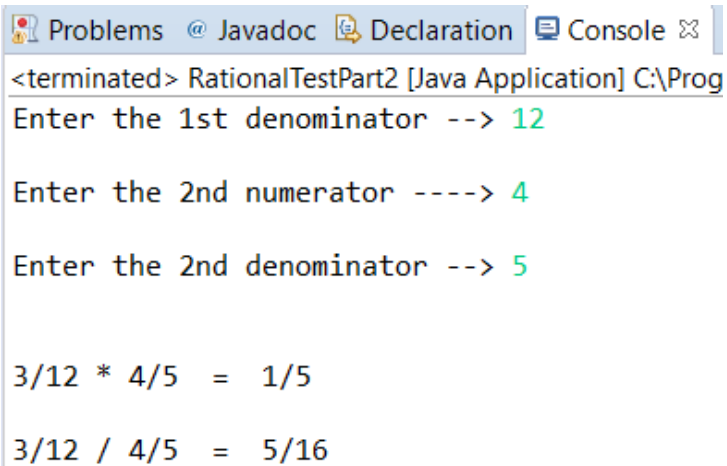
and reduces to 3/4

For the 90 and 100 point versions you will be testing your file with the RationalPart2Test.java file

90 Point Version

The 90 point method adds 2 new object methods: multiply() and divide(). The test file already defines a third Rational object that you are to use. Each method should accept two Rational objects as parameters and multiply/divide the first parameter by the second parameter. Be sure to reduce your answers.

90 Point Version Output



```
<terminated> RationalTestPart2 [Java Application] C:\Prog
Enter the 1st denominator --> 12

Enter the 2nd numerator ----> 4

Enter the 2nd denominator --> 5

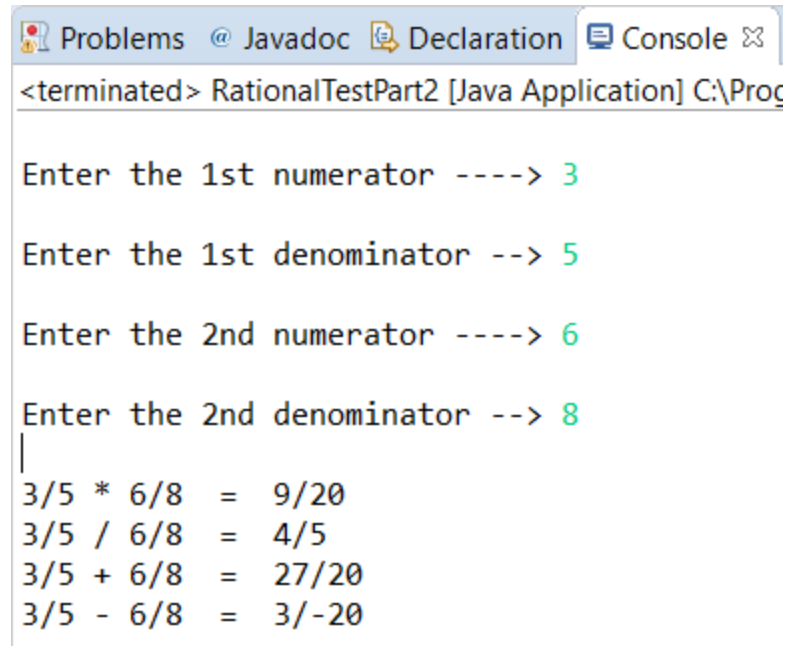
3/12 * 4/5  =  1/5

3/12 / 4/5  =  5/16
```

100 Point Version

The 100 point method adds 2 new object methods: `add()` and `subtract()`. The test file already defines a third `Rational` object that you are to use. Each method should accept two `Rational` objects as parameters and add/subtract the first parameter by the second parameter. Be sure to reduce your answers. **You will need to take the comments out of the test file to run the add and subtract methods.**

100 Point Version Output



```
<terminated> RationalTestPart2 [Java Application] C:\Pro...

Enter the 1st numerator ----> 3

Enter the 1st denominator --> 5

Enter the 2nd numerator ----> 6

Enter the 2nd denominator --> 8

|
3/5 * 6/8 = 9/20
3/5 / 6/8 = 4/5
3/5 + 6/8 = 27/20
3/5 - 6/8 = 3/-20
```

* Don't be concerned with where the negative (-) is when working with a negative rational value.