**Summary**

In this assignment, you will build a program using techniques from previous CSS courses that counts unique fractions (that are positive). No new constructs from our readings are expected, and as such, you can complete this program with just a handful of lines (20–30) of executable code in just one main method. You will need to use loops, ifs, comments, etc., (in other words, **structured programming**) to complete this assignment and achieve the outcomes listed below. Included below is a sample execution of your program as well as a list of outcomes your program is to implement; you do not have to report the fractions in reduced form, though you may choose to do so (you do, however, have to count all equivalent fractions, e.g., 12/6 and 6/3 are equivalent to each other and each contribute to the number of occurrences of the reduced unique fraction 2/1).

**Introduction**

Your project is to read in a series of fractions from a text file, which will have each line formatted as follows: “A/B”.  Both A and B are whole numbers.  A **sample** text file is listed below, and the purpose of your program is to read in each fraction and count the number of occurrences for the current fraction.  When all the input is consumed (or as the input is consumed), your program will print out its list of unique fraction and their corresponding count — see the output below.  (You may assume no blank lines or misleading characters.  You may also assume the fractions in the input file are "normal" for a positive fraction, i.e., no division by zero, negative values, both numerator and denominator are integers, etc.  For the Fractions V2 assignment, you will *not* be able to make such assumptions about the input file.)  See the text file link for **one** of the actual inputs I’ll use when testing your submission; I may use other, similarly formatted input files.

**Sample Input File**

Here's a small input file you can test your program on: [fractions.txtPreview the document](https://canvas.uw.edu/courses/1331881/files/58017705/download?wrap=1)

Let's explain how your program should work by considering an even smaller file than the one above:

6/3

7/3

6/3

12/6

If your program read that file, it should return the following console output:

6/3 has a count of 3

7/3 has a count of 1

Note that both 6/3 and 12/6 are counted as occurrences of the same fraction. Note too you do not, for this assignment, have to report the counts for the fractions in simplest form (e.g., 2/1). You will have to do the fraction reduction for Fractions V2, however.

**Questions**

Answer the following questions using multi-line comments in your code:

* Can you complete this without using arrays? What is the least number of variables you can use to solve this problem?
* Can you use just one array to solve this? What would the data type be of that array?
* What does it mean to make a class so another class is inside (or part of) the first class, so that it is composed of other data types?  What does "composition" mean in that case?  How is it done?
* What are some solutions to the reduction problem other than Euclid's GCD (greatest common divisor) algorithm? (You may want to look up GCD online.)

**Notes and More Requirements**

* Name your main Java file FractionsV1.java.  Most people will have only a single class and thus will upload only a single file, this Java source code file.  You may, if you wish, solve this problem using multiple classes, in which case you'll submit multiple files.  But your main program should still be in a file called FractionsV1.java.
* Your code should correctly specify the input file as fractions.txt, **without** an absolute path or any other environment-specific specifications.  The program should **not** ask for user input from the console.
* Follow the [Coding Style Guidelines](https://canvas.uw.edu/courses/1331881/pages/coding-style-guidelines)!
* Be sure to put code that gets lengthy into its own method!
* Remember to comment your code.  **Every method (including main) must have a Javadoc comment.  A Javadoc comment must also be given at the top of the file (describing the class).** Method comments describe what goes in/out of the method and what the method does.  Include comments inside methods/classes as needed.
* This assignment is particularly detailed in its guidance in terms of outcomes; expect future assignments to test your ability to build your own ***programming contract*** for the problem at hand.
* Test your code as you add features (either methods or variables) incrementally, making sure your small change works as you envisioned it to.
* Try to keep in mind basic “structured programming” principles such as modular design (ie, decompose complex tasks into individual methods), good variable & methods names (self-documenting code), and comments.
* **NB:  By "unique fraction" I do not mean "unique string value" but "unique rational number value."**
* Consider reading one line at a time (with scanner.nextLine()), and using the [split (Links to an external site.)](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#split%28java.lang.String%29) function (defined in class String) to give you an array with two elements in it (the numerator and denominator, respectively).
* Your program must work with the sample input file.  Your program must also work with input files besides the sample given, even long files (e.g., 100,000 lines long).
* Don’t wait till the last minute to get help from the instructor or our QSC tutors!

**What to Submit**

Please **submit your file inside a zip archive called FractionsV1.zip,** even though there will be only one file in it (you do not have to include the input file).  If you are not sure how to create a zip archive, do a Google search on "create zip archive windows" (swap out your computer's operating system, as needed).  Make sure you give yourself enough time before the due time to learn about and create the archive!

**About This Document**

Original assignment by Rob Nash, Autumn 2014. Minor edits and additions by Johnny Lin, August 2015-September 2019.

**Rubric**

Fractions V1 Rubric (2)

| Fractions V1 Rubric (2) | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning Outcome Questions answered in comments |  | 1.5 pts |
| This criterion is linked to a Learning Outcome Correctly counts fractions |  | 10.5 pts |
| This criterion is linked to a Learning Outcome Comments in code |  | 3.0 pts |
| This criterion is linked to a Learning Outcome Code Compiles |  | 5.0 pts |
| This criterion is linked to a Learning Outcome Code Runs |  | 5.0 pts |
| Total Points: 25.0 | | | |