1. Understanding the problem:

For this assignment, I think what we are supposed to do is to create a class called rationales and have two objects that fit this class. Each object should have a numerator and a denominator, as well as overloaded functions to print them and to add, divide, subtract, and multiply them. The user should input the expression in the required formatting (spaces between the fractions and the operator) and the program should be able to read this format, separate the fractions and the operator, and perform the function.

2. Reflection:

To check my answers, I used a calculator to make sure the output is correct. I tested mainly 2 different things: the size of the numbers (so I know I am reading and interpreting the strings correctly) and the negatives (to make sure that the program was using the correct algorithms to deal with negatives). This is because these were the two hardest parts of the program and the things that I had to spend quite a bit of time designing. I didn’t try to plug in letters or to use a different formatting because the instructions explicitly said to expect valid user input. When designing, I used no space for a negative number. I did this because I felt it would be clearer to have operators with spacing and negative numbers without; however, with that space I got varying answers from what I had designed. It took that space as a 0 and assumed the second fraction is 0, and then computed the resulting fraction.

Testing (examples, I did many others but this provides a brief overview):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| What | Why | Input | Output | Correct |
| Basic values | To see if it compiles and comes close to actual results | 3/4 + 3/5 | 27/20 | Yes |
| Different operations | To see if it works for all operations | 4/5 \*/+- 5/20 | 1/5, 16/5, 21/20, 11/20 | Yes |
| Negatives | To see how it handles negative numbers along with subtraction | 7/-8 - -5/6 | -1/24 | Yes |
| Large numbers | To see if it is converting the string to fractions correctly | 325/694 \* 578/312 | 7225/8328 | Yes |