

CSCI 241 Assignment 2

Calculations

Part 1 Due: Tuesday, February 16, 2021 at 11:59 p.m.

Part 2 Due: Thursday, February 18, 2021 at 11:59 p.m.

➔ **NOTE:** This assignment has 2 parts and will take more time than the first one.
Part 1 due two days **BEFORE** Part 2. Don't start Thursday night!!!

Description

In lab this week you have learned to unravel some of Java's error message mysteries. Chapter 2 of the text has also introduced you to basic arithmetic operations. You can put both of those to use this week in Assignment #2. You will correct errors in an already-written program, plus make a new one of your own that does similar work.

Part 1: Splitting a dinner tab

Many of us have gone out to dinner with a friend and then split the bill. Remember those "good old days"? (I miss them! 😞) That includes figuring out a tip (which can typically range from 15% - 20%). (Of course, if you have ever worked as a server, you probably tip generously ;-)

The program you will complete asks the user for various kinds of information related to calculating such a bill for 2 diners, and calculates what the separate bills will be, including either a 15% or 20% tip.

Starting Steps: copy project and fix errors

You will start with a partially-written BlueJ project named **Assign2Calculations**.

- Download the zip file for this project from Canvas, and unzip it.
- Open **BlueJ** and open the **Assign2Calculations** project.

You will see a class named **Dinners**. This program is *supposed to* ask the user for the necessary input information, perform several calculations, and print the results to the terminal window. However, just like in your lab exercises, there are LOTS of errors – misspellings, syntax, logic, lines out of order, lines missing. Your job, of course, is to fix them.

One clue: the comments (designated with `//`) tell you basically what the program should do in each section of code – read them carefully. The code below each comment should accomplish what the comment describes.

Once the program is working correctly, it will not only add and multiply to get the resulting numbers, it will also round the total bills for each person. Each person's bill should include 5.5% tax (the current rate in Kenosha County). The tip is calculated on the bill's amount before adding the tax. The tax is also calculated on the bill's amount (before tip).

We can round a decimal number to 2 decimal places by performing these steps (algorithm easily extendable if you need more decimal places):

1. Take the original number and add 0.005 to it.
2. Multiply this number times 100 (this will put all the digits you wish to keep on the integer side of the decimal point).
3. Use integer typecasting to capture the integer part of the resulting number.
4. Divide this number by 100.0 (which is the same number you multiplied with in step #2) to get 2 digits to the right of the decimal place.

For example:

$$23.624 + 0.005 = 23.629$$

$$23.629 * 100 = 2362.9$$

$$(\text{int}) 2362.9 = 2362$$

$$2362/100.0 = 23.62 \text{ (which is 23.624, rounded)}$$

$$23.627 + 0.005 = 23.632$$

$$23.632 * 100 = 2363.2$$

$$(\text{int}) 2363.2 = 2363$$

$$2363/100.0 = 23.63 \text{ (which is 23.627, rounded)}$$

Here is an example of what should appear in the terminal window once it works, using specified input values for testing. Note that the dollar amounts might not always contain 2 decimal places; don't worry about it for now, we'll learn how to take care of that later. Most of what you see in the box is printed by the program, but I have specified data entered from the keyboard in a **bold and underlined** font to distinguish it from what the program prints:

```
Welcome!  This program will help you split your dinner check for 2.
It will ask you for the prices of all dinner items, one at a time.
At the end, it will show the cost of each meal with 15% or 20% tip.
```

```
Enter today's month number: 2
```

```
Enter the number of today's day: 9
```

```
Enter cost of appetizer (enter 0 if none): 7.50
```

```
Cost of appetizer per person = $3.75
```

```
Enter price of drink #1: 3.00
```

```
Enter price of drink #2: 4.00
```

```
Enter price of entrée #1: 9.95
```

```
Enter price of entrée #2: 12.50
```

```
Enter price of dessert #1 (type 0 if no dessert ordered): 0
```

```
Enter price of dessert #2 (type 0 if no dessert ordered): 5.25
```

```
Dinner out for 2 on 2/9:
```

```
First person's total (with tax and tip) is either $20.12 or $20.96
```

```
Second person's total (with tax and tip) is either $30.73 or $32.0
```

As you work on the assignment, test other numbers as well and see that your program calculates the correct results.

We will assume the user enters decimal or whole number values when expected since we don't know how to check for data input errors yet.

Finish this part and submit your entire BlueJ project in your Computer Science lab account, and submit the `Dinners.java` file in Canvas, by Tuesday, February 16 at 11:59 p.m.

You will resubmit it once you have finished the second part, by Thursday, February 18 at 11:59 p.m.

Part 2: Writing a New Program from Scratch

In the second part of this assignment, you will create a class that does other calculations from numbers entered from the keyboard. This program takes integer input for the numerator and denominator of 2 fractions, performs 2 calculations with the fractions, and prints the results in a neat format.

Follow these steps in order for part 2 of Assignment #2:

- a. Add a new class named **Fractions** to this project.
- b. Fill in the proper heading comments, including a class description of your own choosing. Remember the standard heading specified in the syllabus:

```
/**
 * Name: John Doe (doej)
 * Course: CSCI 241 - Computer Science I
 * Assignment: 2
 *
 * Project/Class Description:
 * (summarize here)
 *
 * Known bugs:
 * (write the word none, or describe)
 */
```

Don't forget the "Known Bugs" line!

- c. Add a **main()** method to your class. The program will ask for each part of 2 fractions (first numerator, then denominator), and will calculate and print the values from adding the 2 fractions and then multiplying the 2 fractions (see sample output on next page).
- d. Think about an algorithm that will guide you toward accomplishing your task. What needs to be done in your program, and in what sequence? Which arithmetic operators will you need, and what kind of data types will you need for your variables?

Here is a basic algorithm for your program:

- Explain the program's purpose to the user.
- Ask the user for integers for the numerator and denominator of each of the two fractions.
- Remember this formula to calculate the *sum* of two fractions:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

- Remember this formula to calculate the *product* of two fractions:

$$\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$$

- Include your algorithm steps as comments inside your **main()** method.
- Add an `import` statement, if you need one.
- Write the code that fills out the actions described by your algorithmic comments.
- Test your program, fixing anything necessary along the way.

There is a sample run of the program in the box below. *Match this as closely as you can in your own program! When I test your program, I'll expect to enter data in the sequence seen below.*

Once again, here is an example of what should appear in the terminal window once it works (data entered from keyboard is shown as **bold and underlined** to distinguish it from what the program prints):

```
Enter numerator 1 integer: 1
Enter denominator 1 integer: 4
Enter numerator 2 integer: 3
Enter denominator 2 integer: 8

Fraction 1:  $\frac{1}{4} = 0.25$ 

Fraction 2:  $\frac{3}{8} = 0.375$ 

 $\frac{1}{4} + \frac{3}{8} = \frac{20}{32} = 0.625$ 

 $\frac{1}{4} \times \frac{3}{8} = \frac{3}{32} = 0.09375$ 
```

We will assume the user reads directions carefully and enters integers when asked. Again, since we don't yet know how to control exact spacing and decimal places, don't worry about that.

Submission Requirements

Computer Science Lab project submission: Submit the completed implementation for the `Assign2Calculations` project, which includes your `Dinners.java` and `Fractions.java` files. Submit the BlueJ project from the main BlueJ project window *from your Computer Science lab account*. This will send a copy of your completed implementations over to be graded. Remember, you can send them more than once if you need to make changes.

Canvas: Submit your `Dinners.java` and `Fractions.java` files to Canvas so I can make notes on them for you to see as feedback.

Grading Criteria

Your program must compile without errors to be accepted. All of the following criteria will be used to evaluate your program:

Correct submission of project	3 pts
<code>Dinners.java</code> program submitted by 11:59 p.m. on 2/16/21	2 pts
Corrected <code>Dinners.java</code> program (submitted by 2/16/18)	7 pts
Fractions: Appropriate and complete comments	5 pts
Fractions: Correct program behavior	5 pts
Fractions: Follows style conventions (identifier names, indentation, etc.)	3 pts
Total points:	25 pts