

Computer Science 145

Lab 2 (10 points)

Due at 11:59 pm the night before your scheduled lab time.

Read all of the instructions. Late work will not be accepted.

Introduction

You should have enough time to complete this lab during the lab session. In your Git repository for labs, create a directory named *lab2*, place your submission in this directory, and push it to the remote GitHub repository before the deadline. If you have questions, be sure to ask the TA. Ask questions often. Labs are your opportunity to get personalized help!

This lab is to be done on the Linux operating system. You must use Linux to complete this lab – basic knowledge of Linux is one of the course outcomes. If required, you can remotely access the campus Linux lab computers (see <https://access.cs.wvu.edu/>).

Lab Rules

You are encouraged to collaborate with your peers when completing the labs, but certain limitations apply. You may discuss the lab with a partner and work together to learn how to write Java code. However, the Java code you submit to GitHub must be your own work. You are required to write the code independently without assistance from others.

Operands and Operators

Assume that you have a keyboard on which the period key is broken, however, you still want to perform a numerical calculation with numbers that have a single decimal number. For example, you want to calculate 20.4×17.7 , but can only enter 204 and 177. The desired result is 361.08. To demonstrate calculating 20.4×17.7 on a “broken” calculator, assume the values 204 and 177 were input. How can you convert them to 20.4 and 17.7? By using the modular and integer division operators! For example, 204 modulus 10 has a remainder of 4, which gives the decimal value .4, while 204 integer division 10 gives 20, which is the number before the decimal in 20.4.

Hint: Assume that for the input values 204 and 177 you have successfully “extracted” the whole and decimal values 20, 4, 17, and 7. How can you “perform” the calculation 20.4×17.7 ? Observe that multiplication of decimal values requires you to multiply the integer of the first number by the integer of the second number, and add that result to the product of the integer of the first number by the decimal of the second number, and add that result to the product of the integer of the second number and the decimal of the first number, added to the product of the decimal in the first number by the decimal of the second number:

```
20.4 * 17.7
(20 * 17) + (20 * 7 * 0.1) + (17 * 4 * 0.1) + (4 * 7 * 0.01)
340 + 14 + 6.8 + 0.28
361.08
```

1. Download from the course website BrokenCalculator.java and save it to your Lab2 folder.
2. That file is incomplete and is a skeleton of a sample program. You are welcome to use it or write your own. Only two lines of code have been written, which you are not allowed to edit. The rest are comments. Lines of code that say ‘// COMPLETE THE CODE’ you will need to write. Read the comments for each section to get a sense of what code you need to write. Also, the number of ‘// COMPLETE THE CODE’ comments in that file is how many lines of Java code were written for the solution. You might need fewer lines of code or more.

For the lines of code that you write, you are allowed to ONLY use the assignment operator, and you can ONLY use the following mathematical operators:

- /
- %
- *
- +

You are NOT allowed to cast variables from one type to another. Only integers.

There are 8 parts to the code, labeled A through H. If you choose to keep that specific skeleton of a program, use the below hints:

- A: Use the code above to get two inputs from the command line.
- B: This requires a single use of the System.out.println method.
- C: Use only the / and % operators. Follow the logic in the description above.
- D: This requires a single use of the System.out.println method.
- E: Do the same for the second integer as you did for the first integer (step C). Use only the / and
- F: The same as step C, but for the second integer.
- G: Use the Hint above for an explanation on how to do this.
- H: This requires a single use of the System.out.println method.

In your Java code, you CAN make use of periods (for floating-point numbers), but when the program is RUN, the user CAN ONLY enter integer (non-decimal) numbers. A sample output for the completed code is provided. Note that your program should work for ANY numerical input supplied by the user and not just the values 204 and 177.

Validating Output

I have provided a sample output in the assignment description called example_output.txt. You will use this to verify if your program is working correctly.

When you run a program in the terminal, the output is usually displayed directly in the terminal window. However, sometimes it's useful to save the output to a file so you can use it later or share it with others. You can do this using the ">" symbol, which redirects the output of a command to a file. Here's how you can use ">" to save the output of your program to a file:

```
java BrokenCalculator 204 177 > output.txt
```

This will run "BrokenCalculator.java" and save the output to a file called "output.txt".

Once you have the output saved to a file, you can use the "diff" command to compare it to an example output and verify that your program is working correctly. Here's how you can use "diff" to compare the contents of two files:

```
diff output.txt example_output.txt
```

This will compare the contents of "output.txt" and "example_output.txt" and display any differences between them. If there are no differences, that means your program is working correctly and producing the expected output.

Keep in mind that the output of your program may contain whitespace or other characters that are not immediately visible in the terminal window. If you're having trouble getting "diff" to match your output to the example output, try opening both files in a text editor and comparing them visually to see if there are any subtle differences.

Submission

Once you have finished all of the above, add, commit and push `BrokenCalculator.java` to your course repository main branch. Note that this file should be directly within a `lab2` subdirectory of your repository (spelling, spacing, and capitalization matter).

Rubric

Criteria	Points
Top of BrokenCalculator.java contains comments, including your name	1
The program does not contain any syntax errors and runs as intended	7
You submitted your work correctly to GitHub and verified it is visible	2
Total	10

Acknowledgments

Thanks are owed to Filip Jagodzinski and several past TAs for contributions to the lab on which this lab is modeled.