Previous Next

HW02

Bookmark this page

Problem Description

Hello, and welcome! Please make sure to read all parts carefully.

For homework 02, you will be creating a calculator that can perform the following operations: add, subtract, multiply, divide, and alphabetize. The operations work as follows:

- `add` - takes two integers, adds them together and prints out the result
- `subtract` - takes two integers, subtracts the second entered integer from the first and prints out the result
- `multiply` - takes two doubles, multiplies them together and prints out the result to two decimal places
- `divide` - takes two doubles, divides the first entered double by the second and prints out the result to two decimal places
- `alphabetize` - takes two words of only letters, and tells which word comes before the other lexicographically

Solution Description

Name your program `Calculator.java`. It should work as follows:

1. Print out the list of operations for the user.
2. Prompt the user to enter an operation. This operation must be processed as case-insensitive.
    - If the user enters an invalid operation, the program should print the following error message and terminate gracefully.
      `Invalid input entered. Terminating...`
3. Perform the chosen operation and print the correct output.
    - If the user is performing an add/subtract operation, prompt the user to enter two integers.
    - If the user is performing a multiply/divide operation, prompt the user to enter two doubles.
    - If the user is performing an alphabetize operation, prompt the user to enter two words.
    - If the user inputs an invalid type (e.g. inputs doubles for add/subtract) for the given operation, print the same error message shown above.
4. The program should terminate gracefully after the result of the operation is printed.

Note that 0 must not be the divisor if you are dividing. If a 0 is the divisor, print the same error message shown above and terminate your program. For the multiply and divide operations, format your output using `printf`.

Your program must also have at least one switch statement and one if/else statement.

For the alphabetize operation, you will be using a pre-defined String method that compares two strings lexicographically, and returns an integer depending on which String is larger. If a 0 is returned, both Strings are lexicographically equal. A positive integer is returned if the first string is lexicographically greater than the second string, or else the result would be negative. See the example output for how the result should be printed to the user for the cases where Strings are equal.

When prompting for integers, doubles, or Strings, spaces will serve as dividers between the two inputs. Remember that when performing multiply/divide operations on doubles, the answer should only include two numbers after the decimal point.

Note: For the autograder to run properly, you must instantiate only 1 Scanner object within the main method, outside of any loops or conditionals. To guarantee this, your very first statement of main should be of the form: **Scanner <name> = new Scanner(...);** and there should be no other statements of the same form in the rest of the program.

Example Outputs

User input is **bolded**. Please make sure to follow the exact formatting as shown below.

```
List of operations: add subtract multiply divide alphabetize
Enter an operation:
add
Enter two integers:
3 4
Answer: 7

List of operations: add subtract multiply divide alphabetize
Enter an operation:
Subtract
Enter two integers:
5 8
Answer: -3

List of operations: add subtract multiply divide alphabetize
Enter an operation:
mulTIply
Enter two doubles:
3.561 7.63
Answer: 27.17

List of operations: add subtract multiply divide alphabetize
Enter an operation:
divide
Enter two doubles:
12 3
Answer: 4.00

List of operations: add subtract multiply divide alphabetize
Enter an operation:
alphabetIZE
Enter two words:
Zebra apples
Answer: apples comes before Zebra alphabetically.

List of operations: add subtract multiply divide alphabetize
Enter an operation:
alpHabEtize
Enter two words:
hello Hello
Answer: Chicken or Egg.

List of operations: add subtract multiply divide alphabetize
Enter an operation:
add
Enter two integers:
1.23 2.34
Invalid input entered. Terminating...

List of operations: add subtract multiply divide alphabetize
Enter an operation:
modulus
Invalid input entered. Terminating...
```

Allowed Imports

To prevent trivialization of the assignment, you may only import `java.util.Scanner`.

Feature Restrictions

There are a few features and methods in Java that overly simplify the concepts we are trying to teach or break our auto grader. For that reason, do not use any of the following in your final submission:

- `var` (the reserved keyword)
- `System.exit`

Grading

Homeworks are graded in an "all or nothing" manner. If your code is correct, you receive a 100 for the assignment; if it isn't, you receive a 0. Along those lines, this program is an exercise in following specifications. Make sure that the output from your program solution match the **exact format** of the expected output. If not, you will receive errors from Vocareum. Also, as mentioned earlier in this course, you should use your local editor and JDK to compose and comple/run/debug your code respectively. Any errors you may then receive from Vocareum will likely be a mismatch in the assignment specification and your code -- e.g. a having a missing or extra space in your output or not rounding numbers as expected.

Allowed Collaboration

When completing homeworks for CS1331 you may talk with other students about:

- What general strategies or algorithms you used to solve problems in the homeworks
- Parts of the homework you are unsure of and need more explanation
- Online resources that helped you find a solution
- Key course concepts and Java language features used in your solution

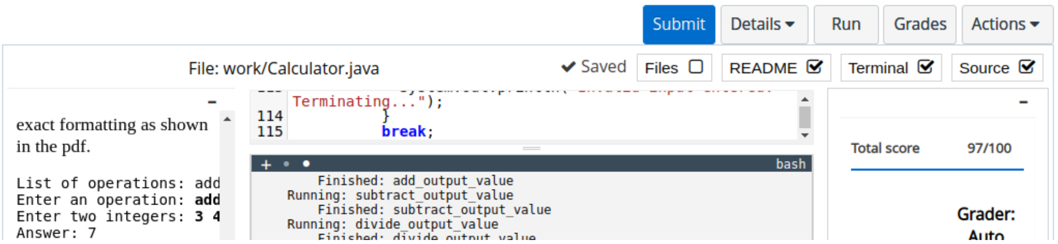
You may **not** discuss, show, or share by other means the specifics of your code, including screenshots, file sharing, or showing someone else the code on your computer, or use code shared by others.

The Vocareum (code editor) interface has six main components:

- The **Drop-Down** in the top left. This lets you choose from multiple available files. Note that this drop-down will only be visible in assignments that require multiple files.
- The **Build / Run** button. For all assignments in this course, the build and run button will perform the same action: compile your code and run a file scan. Building and running your code will not count towards your total allowed submission attempts, therefore you are free to build / run as many times as needed.
- The **Submit** button. This will compile your code, run a file scan, grade your assignment, and output results to console. Note that for most assignments in this class, you will only be allowed a limited number of submissions. A submission is counted when the submit button is clicked, regardless of whether or not your code is able to compile or if there are any file issues. Therefore, **we highly recommend that you build or run your code before submitting to ensure that there are no issues that will prevent your code from being graded and that every submission attempt will generate meaningful results**.
- The **Reset** button. This will revert all your changes and reset your code to the default code template.
- The **Code Window**. This is where you will write your code. Again, We highly recommend copying the starter code and working in your preferred IDE.
- The **Output Window**. This window will appear whenever you build, run, or submit your code and will display the results for you to view.

For additional help, please visit the Vocareum information page located in the course information module!

HW02 (External resource) (100.0 points possible)



exact formatting as shown in the pdf.

```
List of operations: add
Enter an operation: add
Enter two integers: 3 4
Answer: 7
```

```
Terminating...");
            }
            break;
```

```
        Finished: add_output_value
    Running: subtract_output_value
        Finished: subtract_output_value
    Running: divide_output_value
        Finished: divide_output_value
```

```
List of operations: add
Enter an operation: sub
Enter two integers: 5 8
Answer: -3


List of operations: add
Enter an operation: mul
Enter two doubles: 3.56
Answer: 27.17

List of operations: add
Enter an operation: div
Enter two doubles: 12.0
Answer: 4.00

List of operations: add
Enter an operation:` al
Enter two Strings:` "He
Answer: Hello comes bef
```

Rubric

```
        Finished: divide_output_value
    Running: divide_prompt
        Finished: divide_prompt
    Running: add_prompt
        Finished: add_prompt
    Running: add_op_case_sensitivity
    Testing Calculator with operation aDd
    Testing Calculator with operation ADD
    Testing Calculator with operation add
    Testing Calculator with operation adD
    Testing Calculator with operation AdD
        Finished: add_op_case_sensitivity
    Test run: 35, Passed: 34, Failed: 1
    Score: 97/100                    ^C
ddd_v1_w_Lkp_1281951@runweb57583:~$
```

Grader	
calc_score	97/100