Objective

The purpose of this assignment is to test your proficiency with Java loops and methods. It will also serve to check if you can integrate older concepts with newer ones.

For this assignment you are required to write one program with several methods. Please email your file "Evaluate.java" to jayarama@cs.fsu.edu

Problem - Simple Calculator

In the distant and slightly dystopian future, the Grand Poobah has banned the use of hand held calculators. Some say he was hit on the head with one. Others believe it is due to his fear of accountants. Either way, hand calculators are a thing of the past. However, you find yourself in sudden need of a calculator that can not only perform simple calculations, but can also find the number of words in a sentence. No worries. You are a Java programmer and you can write a console application to serve both purposes.

This program does not serve any practical purpose and is, in fact, a very inefficient way to design a calculator. This exercise is just to give you some practice in writing methods.

Specifications

• Create a class with the appropriate name. (5 points)

• This program makes use of methods. The main method should not contain more than 5 lines of code (5 statements).

• In the main method, read a sentence through user input. Then, make calls to the evaluate method two times, once with a String parameter and once with no parameters. (10 points)

• Write 5 methods to perform the 5 basic arithmetic operations: +, -, *, / and %. Give them some sensible name (for example, the method that performs addition can be called add). Each of the methods will take 2 integer arguments and return an integer value. (30 points)

• Write a method called evaluate, that does not take any arguments and does not return a value. This method will read in an arbitrary number of values from the user: an integer followed by a character that denotes an operation. Once you have the number and the operation, call the appropriate method from the five that you just wrote to keep a running evaluated total. The user input ends when the character entered is a ‘?’ . When the user is done with input, print the result. (30 points)

• Write another method called evaluate. This time, it takes a single String parameter. It does not return any value. This method will count the number of words in the sentence and print it. (20 points)

• You may assume that all inputs are valid. You need not test for wrong inputs.

• You can use methods from the String class.

• Please include comments wherever appropriate. (5 points)
Sample Runs

Regular text is what’s printed by your program. Underlined text is user input, shown here as a sample. You will not be printing the underlined parts in your program.

Sample Run 1

Please enter a sentence:
A long time ago, in a galaxy far, far away.
There are 10 words in the sentence.
Enter the expression:
\[
\frac{1}{26} + \frac{2}{19} \div 5?
\]
The result is 3

Sample Run 2

Please enter a sentence:
To thine own self be true, and it must follow, as the night the day, thou canst not then be false to any man.
There are 24 words in the sentence.
Enter the expression:
\[
12 + 9 - 37 + -8?
\]
The result is 63

Generic Guidelines

• Please add your name and FSUID as comments on the top of your program.

• Please make sure you’re only using the concepts already discussed in class. These assignments are used to determine if you passed a certain learning milestone. So, please follow the specifications. Using concepts that are not in the specifications will result in a score of 0 points for that particular specification.

• Please make sure that you’re conforming to output specifications (program name, expected inputs and outputs etc.). Your output must match the sample output exactly (especially the literal text on print statements).

• Please make sure your code is readable and well documented.

• Make sure to compile and run your program before you turn it in. Compilation errors can be costly.

• You can use an IDE for this assignment, but make sure your program compiles and runs on a terminal, since the program will be tested on a terminal.