Description for project

When someone enters equations for the computer to solve, calculators function best.

The user will enter two numbers into this program, and you'll add a total of four mathematical operators: + for addition, - for subtraction, * for multiplication, and / or division. This will allow the program to do computations to verify that the user's input is a numerical string. Ensure that each component is operating properly. Now, we provide extra context so that the user can remain fully informed during the program's runtime.

To help with proper text formatting and feedback, we use string formatters in this process. We can add the remaining operators to the software using the same style as addition in order to give users confirmation about the numbers they are inputting and the operator that is being used with the generated result: If we run the program at this point, the program will execute all of the operations above. However, we want to limit the program to perform one operation at a time. To do this, we will use conditional statements.

Adding Conditional Statements

The goal of the calculator program is for the user to be able to choose among the different operators. Start by adding some information at the top of the program, along with a choice to make, so that the person knows what to do add some conditional statements into the program. Because of how you have structured the program, the if statement will be where the addition is performed, there will be 3 else-if or elif statements for each of the other operators, and the else statement will be put in place to handle an error if the user did not input an operator symbol: This program asks the user to enter an operation symbol before proceeding. Let's utilize the user entering + into Addition as an example. Next, the user enters 5 and 5 when the program asks two digits. The program now displays the solution to the problem and the result: You'll define some functions to manage the user's ability to run the program as often as they like. Put your current code block into a function first. Name the calculation function () the next step is to construct a second function with more conditional statements. You want to provide the user with the option to recalculate or not in this block of code You can model this after the conditional statements seen in calculators, but there will only be one if, one elif, and one else to handle failures in this scenario. Add this method after the def calculate() with the name again(): code preview: Even if the above else phrase handles some errors, it would definitely be clearer if it also accepted Y or N addition to y or n. Add the string function str.upper() to achieve that: You should now add the again() function at the end of the calculate() method to cause the code that prompts the user to choose whether or not to continue to run. You can now calculate as many times as you wish.