### Great Job! 96/100

### Dedust 2 points - 1/x does not work

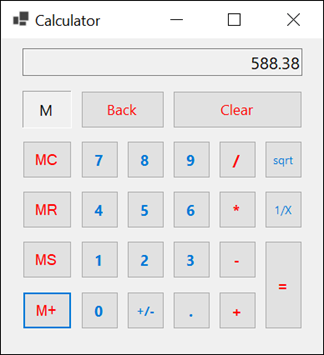
Deduct 2 points – adding a decimal place causes an error.

### **FALL2024 Midterm Project - Create a basic calculator - 100 Points**

For this project, you’ll create a form that lets the user perform the operations provided by a basic calculator. You’ll also create a **Calculator and MemoryCalculator** class that perform the below operations. The instructor will not provide assistance with the code for this assignment, but of course will answer questions regarding the instructions.

Create a **midterm** directory under your main directory of your GitHub repository.

**The Calculator form**



**Operation**

* To perform an addition, subtraction, multiplication, or division operation, the user clicks the first number, followed by the appropriate operator key (+, -, \*, /), followed by the second number and the equals key (=).
* To perform an addition, subtraction, multiplication, or division operation on the result of a previous operation, the user clicks another operator key, followed by another number and the equals key. The user can also repeat the previous operation on the result by clicking the equals keys without first clicking another operator and number.
* To perform a square root or reciprocal operation, or to change the sign of a number, the user clicks the number followed by the appropriate operator key (sqrt, 1/X, +/-).
* To perform a square root or reciprocal operation on the result of a previous operation, the user clicks the appropriate operator key.
* Each time the user clicks a number key, the number is displayed in the text box at the top of the form. This text box also displays the result of an operation when the user clicks the sqrt, 1/X, +/-, or = key.
* To erase the last digit entered, the user clicks the Back key.
* To clear all the values entered, the user clicks the Clear key.
* To clear the contents of memory, the user clicks the MC button. To save the value that’s currently displayed in memory, the user clicks the MS button. To recall the value that’s currently in memory and display it in the calculator, the user clicks the MR button. And to add the value that’s currently displayed to the value in memory, the user clicks the M+ button.
* An M is displayed in the box above the MC button when the memory contains a value.

**Specifications**

* Create a class named **Calculator** that implements the functions of the calculator. Design whatever methods and properties you need for this class.
* If the user tries to divide a number by zero, the calculator should throw an exception that you will catch and display an error message in the text box. The form class should use a try-catch statement to catch a divide-by-zero exception.
* Create a class named **MemoryCalculator** that inherits the **Calculator** class. The **MemoryCalculator** class should add properties and methods as needed to implement the memory function. The private fields of the Calculator class should be made available to the **MemoryCalculator** class as needed.

**Grading**

**Basic Arithmetic Operations (addition, subtraction, multiplication, division)**: 30 points

* This covers implementing the four basic operations and ensuring they work as described.

**Advanced Operations Square Root, Reciprocal, and Sign Change Operations)**: 10 points

* This includes ensuring that additional operations can be performed on the result without needing to start fresh.

**Implemented other Keys (back key, clear key):** 10 points

* The backspace functionality should work properly, allowing the user to delete the last digit entered.
* The Clear key resets all values and clears the display.

**Memory Functions (MC, MS, MR, M+, M display)**: 25 points

* Full functionality of memory buttons, including saving, recalling, and adding to memory, must be implemented correctly, along with displaying the "M" indicator.

**Exception Handling (division by zero)**: 5 points

* The program should correctly handle divide-by-zero errors, using a try-catch block and displaying an appropriate error message.

**Calculator Class Design (methods and properties)**: 10 points

* This evaluates how well the Calculator class is designed, ensuring appropriate methods and properties are implemented for calculator functionality.

**MemoryCalculator Class Design and Inheritance**: 10 points

* This ensures the MemoryCalculator class inherits from Calculator and properly extends its functionality for memory operations, with access to necessary fields.