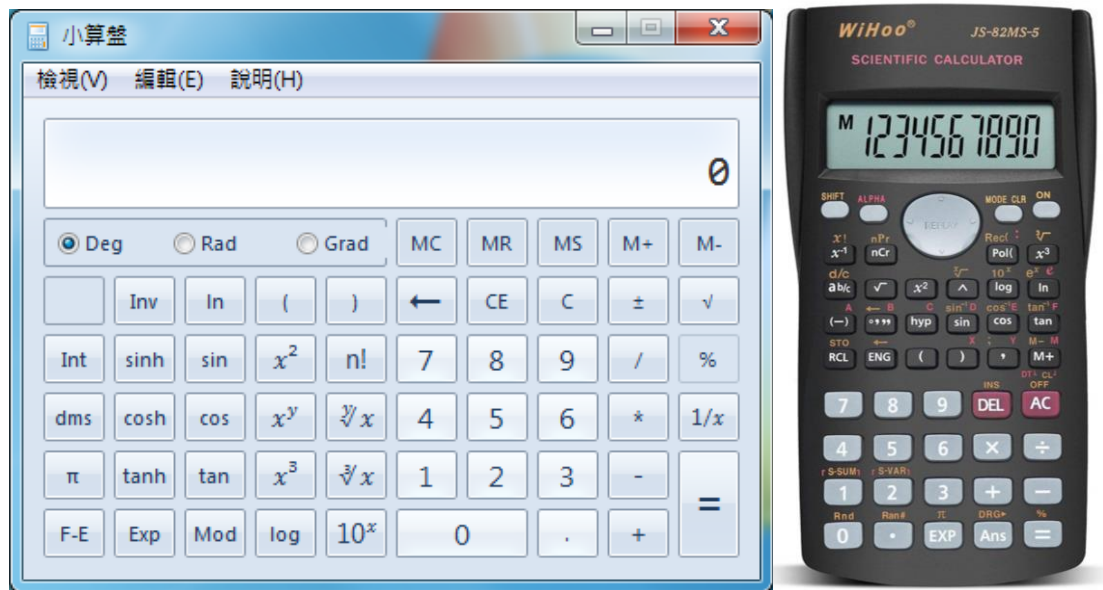


Assignment 003 (2020)

A Full Functioning Scientific Calculator



Create a Windows Form application project and name the solution folder <your ID><yourName>Ass03Calculator. Follow the instructions in the lab to use the **Math** class to implement common calculating functions of a scientific calculator. Try to add more calculation functions to complete the assignment. You can analyze the users' needs in using "小算盤" (before Win 10) or real calculators to provide similar calculation functions. Don't forget to add as many comments as possible to get high grades.

All the mathematical calculation functions are embedded in the **Math** class in .NET Framework. **Math** is a static class (no instantiation is allowed); therefore, you do not need to "new" any object of the **Math** class. You simply use the class name **Math** to access all kinds of mathematic functions, such as Sin(), Exp(), Sqrt(), etc. You should look into the functions provided in the **Math** class to implement various scientific calculations.

Some Tips for your reference:

- **TextBox.Text** or **Label.Text** is a string object. To convert it into a numerical data type, we have learned using static functions of the utility class "**Convert**" to convert a **string** into desired numerical types. For example: **float f = Convert.ToSingle(stringObject);**

`double d = Convert.ToDouble(stringObject), etc..`

- For a mathematic binary expression (calculation), such as +, *, you need to define two operands (data fields of your Form class) to store the values entered by the user and then perform desired binary operation. For unary operation, such as +/-, ln, log, sin, cos, abs, etc., the first operand will be directly used to calculate the final result.
- In addition, a variable should be defined to store the value on the display for future retrieval. This will support the MS (memory store), MR (memory recall), MC (memory clear) functionalities in a common calculator.
- All the methods in the `Math` class are dealing with double-typed data. However, you can use decimal to store a number with a high precision representation.
- The branching statements of C# might be used: `if`, `if-else`, `switch-case` statements. You can define an `enum` to enumerate provided binary operators e.g., '+', '-', '*', or '/'. Or you may define a `char` (or a `string`) variable to store the selected binary operator.
- The primary property of a `RadioButton` or a `CheckBox` is `Checked`, which is a bool data type.
- A set of radio buttons should be grouped within a `GroupBox` to function as a mutual exclusive selection.
- If you like to have an irregular-shaped form, you should provide a button to exit the application.