

University of Utah School of Computing

CS 3500/5010

Handout 2

September 1, 2010

Problem Set One

In this problem set, you will be creating a Visual Studio solution that solves the problem described below. Your solution must be called `ps1` and it must consist of a single project called `calculator`. Use `C#` as your implementation language.

We will retrieve your solution for grading by running the Linux command

```
svn --username cs3500 --password ##### checkout  
    svn://lenny.eng.utah.edu/home/XXXXXXX/cs3500/ps1/trunk ps1
```

(put it all on one line) where `#####` is your grading password and `XXXXXXX` is your CADE login name. We will run the command (which will give us the most recently committed version of your solution) sometime on the morning of September 8. (If you're still working on the morning of September 8, be sure that you don't commit a broken solution.)

You would be wise to verify that the command above works with your repository. If it doesn't work for you, it won't work for us!

Problem

You are to implement a basic four-function calculator. When your project is launched, a window containing the calculator's display and all of its controls should appear. The controls must include the following 17 buttons:

- Ten buttons for the digits 0 through 9.
- Four buttons for addition, subtraction, multiplication, and division.
- A decimal point button.
- An equals button.
- A clear button.

There is a calculator application that comes with Windows 7. Your calculator should behave just like the Windows calculator does *if you put it into scientific mode and use only the 17 buttons described above*. There is one exception. The Windows calculator supports extremely large high-precision numbers. Your calculator should use C# doubles.

Please note that the Windows calculator uses the same precedence rules for operators that are commonly used in programming languages. When simplifying an expression, multiplications and divisions are done first (left to right) followed by additions and subtractions (also left to right).

Documentation Standards

For each source file that contains code that you wrote, include a comment at the top that includes your name, your CADE login name, and your UUID number.

For each class that you implement, include a comment that explains its purpose.

For each method that you implement, include a comment that explains what it does.

For each member variable that you declare, include a comment that explains its purpose.

Grading

Your grade will be based on the quality and the correctness of your code.