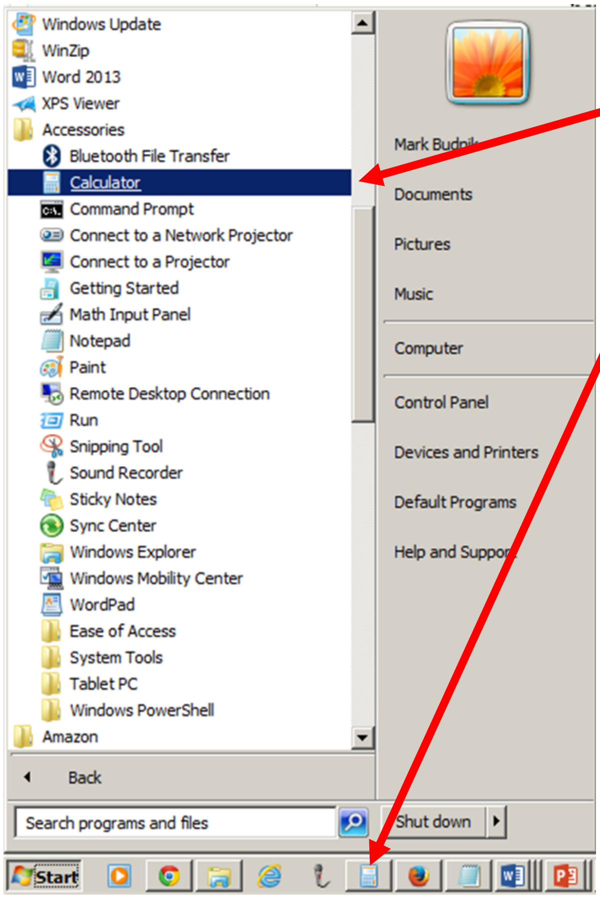
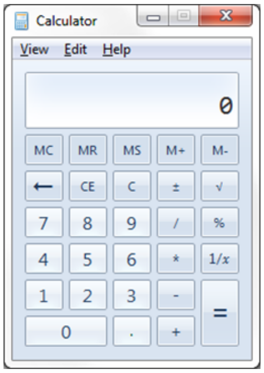
Programming Calculator

1. The Windows Calculator is a very powerful tool for working with binary, decimal and hexadecimal numbers.

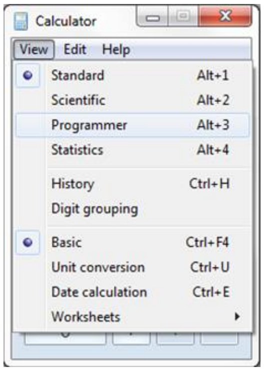
2. Open the Windows Calculator. It is found in the Accessories folder under the Start menu. However, I find the tool so useful that I have added it to the bottom toolbar.



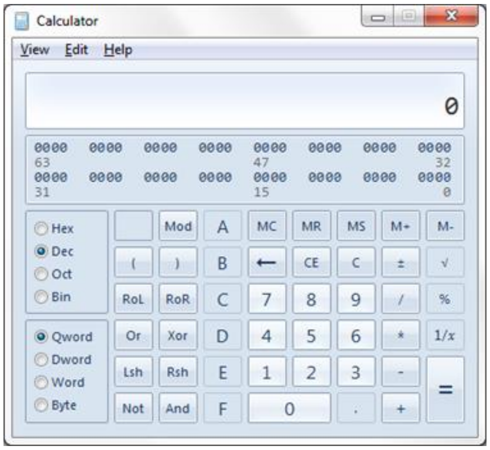
3. When the program opens, the default calculator will probably be shown.



4. Click on the View pull-down menu and select Programmer.



5. This will open the Programmer Calculator view. Note that the Dec(imal) radio button is selected. Any numbers you enter into the calculator at this time will be in base 10. If you look carefully, you may notice that the hexadecimal characters (A, B, C, D, E, and F) are greyed out and cannot be used at this time.



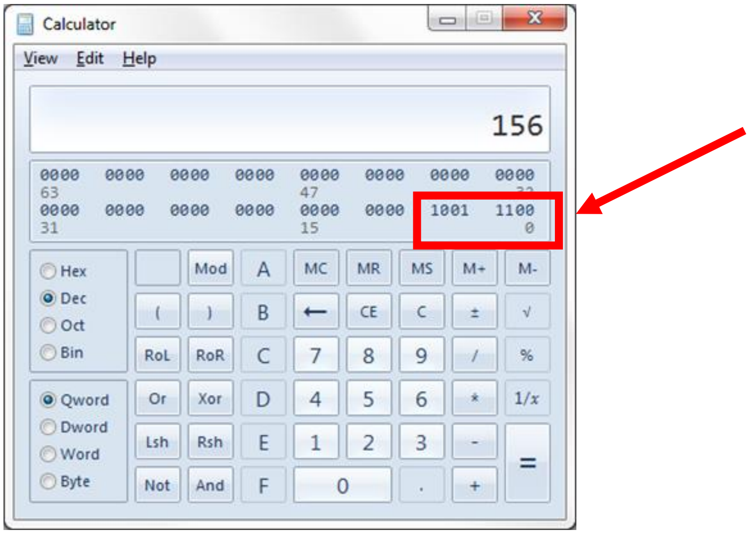
Calculator

```
                                      0
0000 0000 0000 0000 0000 0000 0000 0000
63                  47               32
0000 0000 0000 0000 0000 0000 0000 0000
31                  15                0
```

○ Hex
● Dec
○ Oct
○ Bin

● Qword
○ Dword
○ Word
○ Byte

6. Enter 156 into the calculator. You will notice that the first eight binary digits underneath the main display have also been updated. This shows that 156_{10} = **`1001 1100B`**.



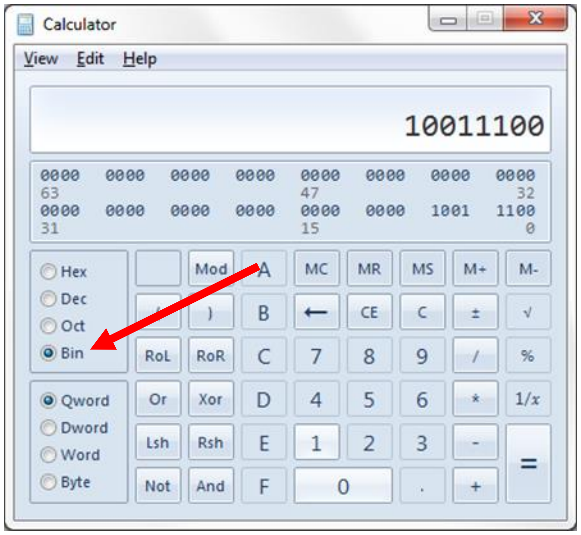
Calculator

```
                                    156
0000 0000 0000 0000 0000 0000 0000 0000
63                  47               32
0000 0000 0000 0000 0000 0000 1001 1100
31                  15                0
```

○ Hex
● Dec
○ Oct
○ Bin

● Qword
○ Dword
○ Word
○ Byte

7. To see this more clearly, you can click the Bin(ary) radio button. **10011100** is now shown on the main calculator display.



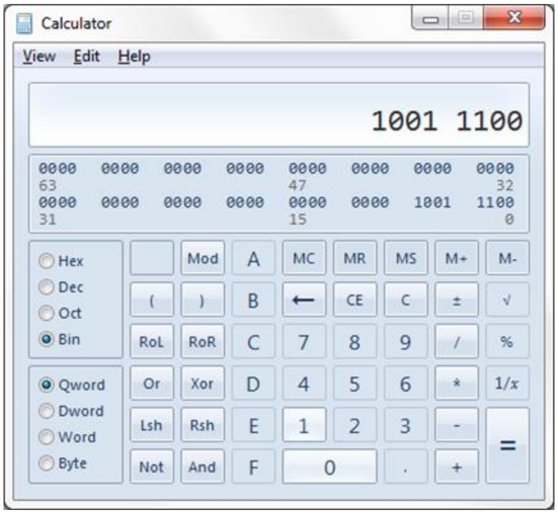
```
10011100
```

```
0000  0000  0000  0000  0000  0000  0000  0000
63                      47                  32
0000  0000  0000  0000  0000  0000  1001  1100
31                      15                   0
```

○ Hex
○ Dec
○ Oct
● Bin

● Qword
○ Dword
○ Word
○ Byte

8. As mentioned in an earlier handout, sometimes, it is easier to group the binary digits in groups of four. To see the display this way, from the View pull-down menu, select Digit grouping.



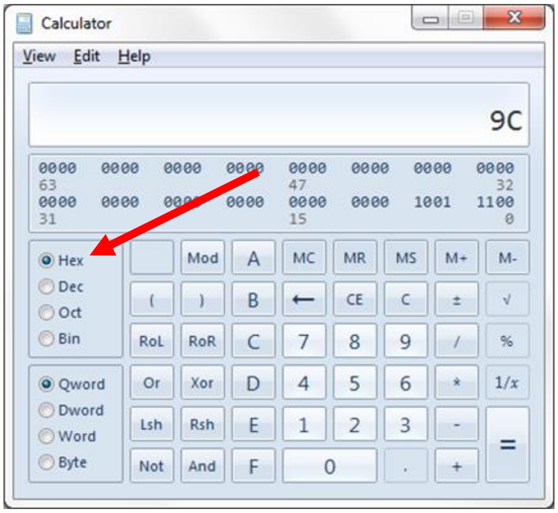
```
1001 1100
```

```
0000  0000  0000  0000  0000  0000  0000  0000
63                      47                  32
0000  0000  0000  0000  0000  0000  1001  1100
31                      15                   0
```

○ Hex
○ Dec
○ Oct
● Bin

● Qword
○ Dword
○ Word
○ Byte

9. To see the number in hexadecimal, click the Hex(adecimal) radio button. Notice, that in Hexadecimal mode, the A, B, C, D, E, and F buttons are no longer greyed out and can be used.



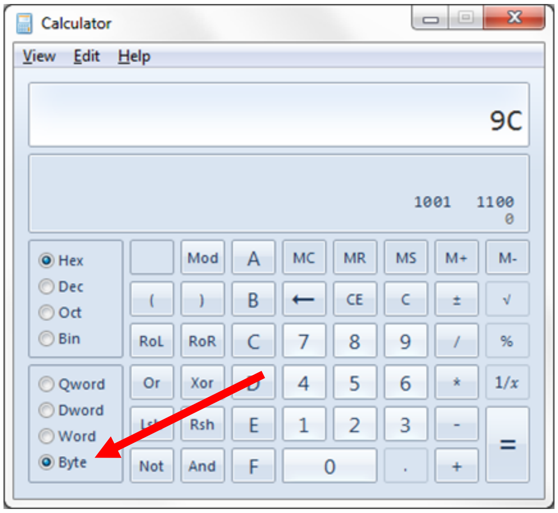
```
                                            9C

0000   0000   0000   0000   0000   0000   0000   0000
63                          47                     32
0000   0000   0000   0000   0000   0000   1001   1100
31                          15                      0
```

● Hex ○ Dec ○ Oct ○ Bin

● Qword ○ Dword ○ Word ○ Byte

10. Finally, in the default Programmer view, the calculator is in Q(uad) Word mode. All of the binary numbers are being manipulated as 64-bit numbers. You can switch to 32-bit (Double Word), 16-bit (Word), or 8-bit (Byte) mode by clicking the appropriate buttons.

    In the example below, the calculator is in Byte mode.



```
                                            9C

                                          1001   1100
                                                    0
```

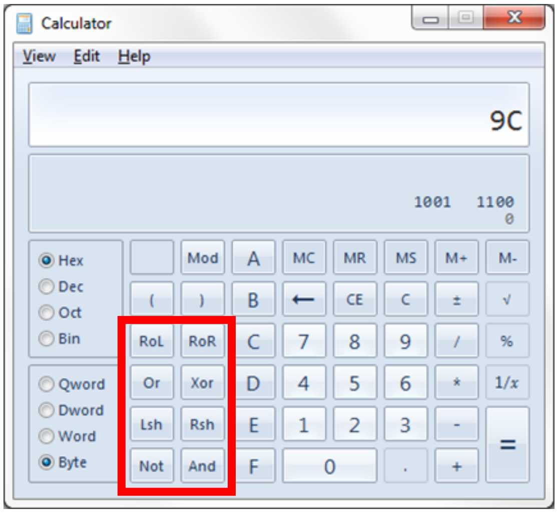
● Hex ○ Dec ○ Oct ○ Bin

○ Qword ○ Dword ○ Word ● Byte

11. You can use the calculator in hexadecimal or binary format just like a traditional calculator. You can add, subtract, multiply (*) or divide (/).

In addition, there are many binary functions you can use like And, Or, Xor, Not, L(eft) sh(ift), R(ight) sh(ift), Ro(tate) L(eft), and Ro(tate) R(right) and a few others.

Our next handouts will introduce a few of these functions if you are not familiar with them.



```
9C

1001   1100
          0
```

● Hex
○ Dec
○ Oct
○ Bin

○ Qword
○ Dword
○ Word
● Byte

All tutorials and software examples included herewith are intended solely for educational purposes. The material is provided in an “as is” condition. Any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for particular purposes are disclaimed.

The software examples are self-contained low-level programs that typically demonstrate a single peripheral function or device feature in a highly concise manner. Therefore, the code may rely on the device's power-on default register values and settings such as the clock configuration and care must be taken when combining code from several examples to avoid potential side effects. Additionally, the tutorials and software examples should not be considered for use in life support devices or systems or mission critical devices or systems.

In no event shall the owner or contributors to the tutorials and software be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.