**Project 3: Bitwise Ops**

**CS 200 • 20+5 Points Total  
Due Friday, February 26, 2016**

**Objectives**

* Create a program that shows the results of various bitwise operations applied to two arbitrary integers. Results should be shown in decimal, hexadecimal, and binary.
* Practice programming in C/C++.
* For extra credit, write your program as a command-line RPN calculator.

**Assignment**

To get some practice with C/C++ *and* gain an understanding of how bitwise operators work, write a program in C or C++ according to the following guidelines:

* Prompt the user for input and read in two integers *a* and *b* from the keyboard.
* Print out the results of each of the following calculations: a & b, a | b, a ^ b, ~a, a >> b, a << b.
* Print each result with an identifying label ("a & b = ") and its value in decimal (base 10), hexadecimal, and binary.
* Make liberal use of Google or other resources to figure out how to do stuff! Also, while printing hexadecimal is a built-in option in C, printing binary is not. Feel free to use algorithms or code you find online for that - just be sure to credit your sources in your project report.
* Pad your numbers with 0s. If you are using 8-bit numbers, 009 in decimal = 09 in hex = 00001001 in binary. Similarly, 127 = 7F = 01111111. Padding will help keep your output neat and easy to read.
* I strongly suggest you use the *unsigned char* data type for your inputs and calculations. *unsigned char* is by default 8 bits long and avoids weird problems with the binary and hexadecimal not matching the decimal due to the sign bit. You can use variables of that type anywhere you use an integer because that’s what it actually is – an integer that is the ASCII code for some actual character.
* For up to +5 extra credit, write your program as an RPN-style calculator. Each number you type is pushed on a *stack*. Every operator you type pops two values off the stack, performs the operation, then pushes the result back on the stack as well as printing it out. For example, entering "3 4 5 + \*" (on separate inputs) prints out 27, since the + adds 4 and 5 to get 9 and then the \* multiplies 9 by 3. Be sure your calculator also handles the bitwise operations; **no extra credit will be given if it only does simple arithmetic**. Here is what the example above might look like in a working program:

Enter a number or operation: 3  
Current top of stack is 3. Continue or ‘Q’ to quit: 4  
Current top of stack is 4. Continue or ‘Q’ to quit: 5  
Current top of stack is 5. Continue or ‘Q’ to quit: +  
Current top of stack is 9. Continue or ‘Q’ to quit: \*  
Current top of stack is 27. Continue or ‘Q’ to quit: E  
Invalid input, please try again. Continue or ‘Q’ to quit: Q  
Your final result was 27. Goodbye.

**Project Report**

The final step of this assignment is to create a report consisting of a cover page, an overview of the project, sample output, and the source code. See Assignment Policies on either the class website or Bb Learn. Just as a reminder, the **cover page** should contain the following information. Correct the date to be the day you actually turn in the project.

Project 3: Bitwise Ops

Your Name  
January 27, 2012

CS 200