

## CS 231 – Lab #5 – Bitwise operators in Java

You have seen how bitwise operators work in assembly language. But in your career, you are much more likely to encounter them in a high-level language. So, this lab is intended to acquaint you with how Java uses bitwise operators.

Create a new folder called lab5 on your USB drive or account, and do all of today's work in there. From the class Web site, you should see a folder called lab5. Download Bitwise.java. You will modify this program.

Bitwise.java is intended to be a simple and interactive program that performs the following bitwise operations:

- |                          |                |
|--------------------------|----------------|
| • Shift operations       | Java operator: |
| ○ Shift left             | _____          |
| ○ Shift right logical    | _____          |
| ○ Shift right arithmetic | _____          |
| • Logical operations     |                |
| ○ AND                    | _____          |
| ○ OR                     | _____          |
| ○ XOR                    | _____          |
| ○ NOT                    | _____          |

As a little puzzle for you, can you figure out which Java operator corresponds to each of the operations named above? Here are your choices, in no particular order:

|      ~      <<      &      >>      ^      >>>

Next, compile and run Bitwise.java. You can see that it is interactive. It begins with some simple specific examples of using bitwise operations. The second part of the program practices the following skills:

- Using the shift-left operator
- Using the shift-right-logical operator
- Using the shift-right-arithmetic operator
- Turning on a specific bit (Assume that bit #0 is on the right end.)
- Turning off a specific bit
- Inverting a specific bit

In each case, the variable `result` is being set to zero. You need to modify these assignment statements, using the appropriate bitwise operators, so that these desired operations are correctly achieved. Look carefully at the binary output for guidance. ✓

Finally, write an expression to accomplish each of these operations in a high-level language such as Java:

1. We have an integer value  $n$ . Assuming that bit #0 is on the right end, how would we invert bit # 12?

2. How would we clear bit # 15? (Clear means to become zero)

3. How would we set bit # 19? (Set means to become one)

4. How would we set the last (i.e. rightmost) 5 bits? ✓