Review: Digital logic

1. What are Boolean values and what are the basic Boolean operations?

2. What is a Boolean expression? Example?

3. Which logical operators correspond directly to the basic Boolean operations in most programming languages?

4. What does this evaluate to: 1 AND (0 OR (NOT(1)))? Write the code in any programming language you know, run and check it, and write the intermediate steps, too.

5. Given the formula or function f(x,y,z) = (x AND y) OR (NOT(x) AND z) for Boolean values x, y, z, write a program for this function in any language and confirm this result:

```
| x | y | z | f(x,y,z) |
|---+---+---+----------|
| 0 | 0 | 0 |        0 |
| 0 | 0 | 1 |        1 |
| 0 | 1 | 0 |        0 |
| 0 | 1 | 1 |        1 |
| 1 | 0 | 0 |        0 |
| 1 | 0 | 1 |        0 |
| 1 | 1 | 0 |        1 |
| 1 | 1 | 1 |        1 |
```

6. Which description of Boolean logic is better - the truth table or the close functional form, and why?

7. How many states does a Boolean function of 5 arguments have?

8. True or false? The expression (x AND y) OR (NOT x AND z) can be represented by *two* different but equivalent Boolean formulas.

9. Do you remember any Boolean algebra laws? List them (on the back side of this test) alongside their definition.

10. Use the Boolean algebra laws to simplify this expression (use the back side of the page): NOT(NOT(x) AND NOT(x OR y)) [Tip: 5 steps].