**Introduction to Coding (in Python)**

**Lesson 2 – Boolean Logic**

**What does Boolean mean?**

Named after George Boole, Boolean logic is based on Boolean variables. A *Boolean variable* is a variable with one of two values: True or False (equivalent to 1 or 0). In python, True and False are reserved key words corresponding to these two possible values. Note that both have the first letter capitalized.

Example 1:

myVar = True

The above example defines a Boolean variable myVar as True.

**Logic 101**

The most important use of Boolean variables is in the evaluation of the ‘truth value’ of statements. This allows the code to do different things based on whether a statement is true or not. The most important operators regarding logic are the comparison operators == and !=, which correspond to “equal” and “not equal”. These can work on numerical and non-numerical data.

Example 2:

a = 1.0

b = 2.0

aEqualb = (a==b)

aNEqualb = a!=b

print(aEqualb)

print(aNEqualb)

The above example prints False and then True, because the variable a is not equal to the variable b. For this example, “a is equal to b” (a==b) is a False statement, and “a is not equal to b” (a!=b) is a True statement. Reading these logic statements out in this manner is an easy way to spot errors in code that uses logic operators.

For numerical variables, there are also comparison operators which say whether a number is greater than or less than another number:

a>b “a is greater than b”

a<b “a is less than b”

a>=b “a is greater than or equal to b”

a<=b “a is less than or equal to b”

**PROBLEM 1:**

**Using the variables a and b from example 1, write code to calculate the truth value of the above statements and display each truth value.**

An important not to these operators is that, except for ==, the other character always comes before the equals sign (!=, <=, >=) because otherwise the code would think you were defining a new variable! As a result, you will get a syntax error with an expression like a=<b.

**If else elif and or**

The title of this section is nonsense, unless you somehow know that these are keywords in python!