Lab 3: More Programming w/ Python

(2 pts) Practice: Numbers, Logical Operators, and Variables

First, discuss how you would solve each expression based on precedence and **hand write your answers** to the following expressions. In addition, we will follow the standard in most programming languages, where **integer arithmetic is a result from using only integers,** and if one operand is a float, then the result is a floating point number. The special operator for integer division, //, is a new feature of Python 3, but this is not typical in most languages. In the following expressions, assume integer division when a divide is between integers, when determining your answers, and **write down whether you would use the // or / operator for your division in each of these answers.**

Evaluate the following expressions:

1. 4 ÷ 3 ÷ 2
2. 4 ÷ 3.0 ÷ 2
3. 1 ÷ 2 × 8.0
4. 5 × 10 ÷ 2 +10 ÷ 5.0
5. 3 + 2 × 4 ÷ 5 + 8 + 2
6. (3 + 2) × 4 ÷ (5 + 8) + 2
7. (3 + 2) × 4 ÷ 5 + (8 + 2)
8. 20.0 ÷ 4 × 2^3
9. 5.5 × 2 + 4 ÷ 2
10. false and true
11. not false
12. true or false
13. not true or false and false or true
14. not ((true or false) and (false or true))
15. not true and false
16. not (true and false)
17. false and not false or false

Using variables:

a = 0.0

b = 1.0

c=True

d=False

1. b – a ÷ 10
2. (b – a) ÷ 10
3. not c or not d

Now, **look up the ASCII character set in a web browser**. Determine **the numbers that correspond to the letters in your first and last name**. This includes a capital letter for your first and last name with a space between them. Write down the numbers for both you and your partner.

(2 pts) Practice Python: Testing your solution

Write Python code, **numbers_and_name.py**, to test your handwritten solutions. Using integer or floating point division, where appropriate, write a python program that prints the values of the above expressions based on the rules for integer arithmetic. Remember, if at least one operand is a floating point number, then the result is a floating point number, otherwise it is integer arithmetic.

For Example:

```
print(4 // 3 // 2)
```

Note: for the True/False expressions make sure to type them capitalized. (ex: True or False)

Now, **write a python program that uses the ordinal values from the ASCII character set that you found for your name in #3** and prints the characters corresponding to the values. Your name should come out with the letters beside each other and spaces between your first and last name.

Compare your handwritten solutions with your Python solutions to see if you got the correct answer. For those answers where your handwritten solution differed from the Python solution, document what caused this error. For example, did your make a typo when entering the number in Python or a logical error when calculating your solution by hand?

(4 pts) Design and Write a Program (Flowchart or Psuedocode)

Now, let's **design a program** that takes the weight (in pounds) and radius (in feet) of a sphere as input and outputs the buoyant force. Use $\gamma = 62.4$ lb/ft^3 as the specific weight of water, and the volume of the sphere is $(4/3)\pi r^3$. The buoyant force can be computed by

$$F_b = V \text{ x } \gamma$$

where F_b is the buoyant force, V is the volume of the submerged object, and γ is the specific weight of the fluid. If F_b is greater than or equal to the weight of the object, then your program will output, **"This sphere will float"**, otherwise it will output, **"This sphere will sink."**

STOP. Exchange Designs and Implement Design (Even if you did this on your own outside of lab, you MUST still trade with someone in lab).

Now, **write the python program, buoyancy.py,** using some else’s design. Was the design explicit enough? Were you able to follow their design exactly and get the right answer or did you have to modify it due to logic errors?

(2 pts) Modify the design

Keep the design you have and modify it. What if you want to continue to find the buoyant force for other spheres depending if the user wants to find more? How would this change the design? Do you need more variables? Where do you need to make more decisions?

After modifying the design, implement the additional functionality using a loop,

Choose the right loop syntax to use!

Python Loops:

```
#both loops print hello 10 times

x=0;

while(x<10):

  print(“hello”);

  x=x+1;


for x in range(10):

  print(“hello”);
```

Return the design back to the original group/individual. **Show them the program** you implemented from their design.

Can the group/individual understand the program easier, since it is their design? Does it match how they would have implemented it based on the design they developed? Describe how it doesn’t match.