

## CS5001: Lab 2. Due on Friday, Sep-22-2023.

**Name(s):** Xujia Qin

**Email(s):** qin.xuj@northeastern.edu

You can work on this lab either individually or in small group of two students. If working in a group, include names of all the students in the submission PDF.

Getting credit for this lab. This lab handout has several empty boxes that prompt you to answer a question. As part of the lab, you are to write the answers to these questions inside the boxes. When you are finished, you should create a PDF and upload it on Canvas. If you don't finish, you have until 11:59 PM on Friday, Sep-22-2023 to submit.

What computer to use? If your primary computer is a laptop, bring it to the lab to work on, as lab is an excellent opportunity to get started with Python on your machine. You should follow the instructions on the course website. Ask a TA for help if you have problems with your installation. If you prefer, you could also use one of the machines in the lab room to work on this lab assignment.

Lab Materials. Lab materials can always be found on Canvas under the appropriate lab posting.

For today's lab, you need this handout (which is also online), plus the file `FormatPlay.py`, which is a text-based Python script.

### 1. Boolean expressions

For each example in this section, write down what you think is the value of the expression. Then use Python in interactive mode to find out if your mental reasoning is OK or not. If not, try to figure out why Python gave the answer that it did. Ask a TA member for clarification if necessary. Leave a paper trail of your mistakes and recoveries in the "notes column" as a reminder not to make the same mistakes in the future!

1.1. Numerical Examples. Assume that x,y and z are initialized as follows

```
>>> x = 1
```

```
>>> y = 5
```

```
>>> z = 10
```

Complete the table below. (24 points. 2 points each cell.)

	Expression	I Think the Value Is	Python Says	Notes
1	<code>x &lt; z</code>	True	True	<p>or: if either of the statement is true, the result is true; if both are false, the result comes out false</p> <p>and: if both of the statement are true, the result is true; if either is false, the result comes out false</p>
2	<code>2*y &gt;= z</code>	True	True	
3	<code>2*y &lt; z</code>	False	False	
4	<code>(x&gt;1) or (z!=7)</code>	True	True	
5	<code>y != (z/2)</code>	False	Text False	
6	<code>(x&gt;0) or ((y&gt;0) and (z&lt;0))</code>	True	True	
7	<code>((x&gt;0) or (y&gt;0)) and (z&lt;0)</code>	False	False	

Recall that if you just enter an expression in Python interactive mode, then Python will display the value of that expression. Thus, something like `>>> x < z` will result in the display of either “True” or “False”.

1.2. String Examples. Assume that x, y, and z are initialized as follows

```
>>> x = 'Cornell'
```

```
>>> y = 'Northeastern'
```

```
>>> z = 'Yale'
```

Now complete this table: (16 points, 2 points each cell)

	Expression	I Think the Value Is	Python Says	Notes
1	<code>x != z</code>	True	True	case sensitive  y[1:] z[1:] begins with 'o', and 'a' correspondingly, so in alphabetical order, 'o' is latter than 'a'.
2	<code>x == 'cornell'</code>	False	False	
3	<code>len(x) &gt; len(y)</code>	False	False	
4	<code>y[1:] &gt; z[1:]</code>	True	True	
5	<code>len(x+z) &gt; len(y)</code>	False	False	

## 2. The If-Else Construction

(a) Assume that s is initialized with a string. Complete the following so that it prints Plural if s ends with 'es' and Not Plural if it does not. Fill in the blank in front of the If statement. (10 points)

```
if s.endswith('es'):
```

```
    print ('Plural')
```

```
else:
```

```
    print ('Not Plural')
```

Try this out in Python interactive mode to see if you are correct. This will involve doing: assign a string ending in "es" to variable s; type in the entirety of the if-else statement you have completed above; and see what the answer is. Then do the same thing, except first assigning to s a string that ends in, say, "ss". (You can actually hit the up-arrow key to recover your previous Python commands, which is convenient.)

(b) Assume that s is initialized with a string of digits, e.g., '12345'. Complete the following (write code in the blank spaces) so that it prints seven times the value of the middle digit if s has odd length. Thus, if s is the string '12945', then 63 should be displayed. If s has even length then it should print 3 times the value of the last two digits. Thus, if s is '1246' then 138 should be printed. **(20 points)**

```
s = input("enter digit = ")
```

```
if len(s) % 2 == 1: # Check if the length of 's' is odd
```

```
    middle_digit = int(s[len(s) / 2]) # Calculate the middle digit (assuming integer conversion)
    result = 7 * middle_digit
```

```
else:
```

```
    last_two_digits = int(s[-2:]) # Calculate the last two digits (assuming integer conversion)
    result = 3 * last_two_digits
```

```
print(result)
```

In working on this problem, you may find it easier to enter your code in a new python program in an editor, so you can make changes more easily. Remember to run your new program in the command shell. Use print statements to discover the sources of any errors.

### 3. Pleasing printing

Another file in the zip archive we provided us is FormatPlay.py. Here it is:

```
# FormatPlay.py
# the CS5001 prof (ad.mishra@northeastern.edu)
# Sep 20, 2023
""" A short script that illustrates formatted print. """
from math import pi
x = 355
y = 113
z = float(x)/float(y)
err = abs(z - pi)

print('\nNumerator   Denominator   Quotient       Error')

print('-----')

print(str(x).rjust(5) + str(y).rjust(5) + str(round(z,15)).rjust(19) + str(round(err,15)).rjust(19))
```

It produces the following output:

```
Numerator   Denominator   Quotient       Error
-----
  355   113  3.141592920353983      2.66764189e-07
```

Modify the last line in FormatPlay.py so that the following output is reproduced:

```
Numerator   Denominator   Quotient       Error
-----
  355         113      3.1415929      2.67e-07
```

Numerator	Denominator	Quotient	Error
355	113	3.1415929	2.67e-07

3a) In other words, center the numbers under the column headings, display the Quotient through seven decimal places, and display the Error with three significant digits. Do this by playing with the format specifications and blanks in the string '%3d %3d %22.15f %10.6e'. What does your new print statement look like? Fill in the blank below. **(10 points)**

```
print '%6d %10d %17.7f %12.2e' % (x,y,z,err)
```

OR

```
print(str(x).rjust(6) + str(y).rjust(11) + str(round(z,7)).rjust(18) + str(round(err,9)).rjust(13))
```

---

Comment: both print statements will get the same format result

Change the first print statement to the following.

```
print ('\n\n\n\nNumerator    Denominator    Quotient        Error')
```

3b) What does \n seem to do? (Fill in the blank below.) **(10 points)**

separate lines or create line breaks in the output.

---

#### 4. If-Elif-Else

Consider the following code:

```
x = int(input('Enter x' ) )
y = int(input('Enter y'))
if 1<=x<=3 and 1<=y<=3:
    print ('A')
elif x>3:
    print ('B')
elif y<1:
    print ('C')
elif y>3:
    print ('D')
else:
    print ('E')
```

4a) Complete the following table: **(10 points, 2 point each)**

x	y	Output
2	2	A
1	0	C
4	1	B
1	5	D
0	3	E
3	0	C