

CSC 280 Lab 1: Command line, Expressions, and Assignments

First Name: _____ last Name _____ AU ID _____

1. If your primary computer is a laptop, bring it to the lab to work on, as lab is an excellent opportunity to start Python on your own machine. You can follow the instructions of [SettingupPython.pdf](#) that we hand out in the first day of class or this link:

http://nw08.american.edu/~bxiao/CSC280_15/SettingupPython.pdf

Please use Python 2.7.

If you don't have a laptop, it is OK to use the computer in the classroom. Open IDEL and learn how to create and save a file in a directory. Make sure at the end of the lab, you transfer what you saved on the computer to your personal disks. As the course goes on, it is important that you have your own code Repository.

2. Next, you need to start up the command shell on your computer you are working on. This would be the Command Prompt on Windows and Terminal on Mac OSX. You can follow the instructions here:

http://nw08.american.edu/~bxiao/CSC280_15/RunPythonFromCommandLine.pdf

After you have started the command shell, type python at the prompt. On my computer, you should see output that looks something like this:

```
Python 2.7.9 |Anaconda 2.2.0 (x86_64)| (default, Dec 15 2014, 10:37:34)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
>>> print 'hello world'
hello world
```

3. Useful shortcuts:

If you press the up-arrow key, you obtain the previous expression that you typed in. Pressing up-arrow repeatedly scrolls through all the expressions that you have typed earlier in the session; if you go too far back, you can press the down-arrow key to get to a later expression. The left and right-arrow keys move the text cursor on a single line. Using all of these keys together, you can take a previously-used expression, modify it, and try it again.

4. Lab Instructions

The following pages have a list of expressions. For each expression, first compute the expression in your head, without Python. Write down what you think the value is in the second column of the table. If you have no idea, write "?".

Next, use Python to compute the same expression. You may find it easier to cut-and-paste from the online version of these instruction. Write down Python's result in the third column.

You should always fill in the second and third column of a row before moving on to the next row. You want to learn from earlier examples before moving on to the next one.

If the two values are different, you should try to figure out why Python gave the answer that it did. Come up with a reasonable explanation and put it in the final column. You are not really being graded on complete correctness in these labs (see below) so make your best guess at what is happening; your answer will help me understand how to better aid you.

Don't be stuck with yourself. Feel free to discuss with your neighbor. When you are finished, please turn in your lab to your instructor.

If you do not finish your lab, you have until the next lab to finish it. Please show your work before you leave. If you are not done, you can show me in the beginning of next lab. Lab is graded with effort, not correctness.

3.1 int and float expressions

| Expression | Expected Value | Calculated Value | Reason for Calculated Value |
|------------|----------------|------------------|-----------------------------|
|------------|----------------|------------------|-----------------------------|

| | | | |
|--------------|--|--|--|
| 2**3 | | | |
| 5+2*5 | | | |
| (5+2)*5 | | | |
| -4 - -4 - -4 | | | |
| 2**2**0 | | | |
| (2**2)**0 | | | |
| 6/4 | | | |
| 6/4.0 | | | |
| 9.0*0.5 | | | |
| 9.0**0.5 | | | |
| 6 % 2 | | | |
| 7 % 2 | | | |
| 6.2 % 4 | | | |

3. 2 Types and Casting.

| Expression | Expected Value | Calculated Value | Reason for Calculated Value |
|------------|----------------|------------------|-----------------------------|
|------------|----------------|------------------|-----------------------------|

| | | | |
|-----------------|--|--|--|
| float(4) | | | |
| int(4) | | | |
| int(5.3) | | | |
| float(int(5.3)) | | | |
| int(-5.3) | | | |
| float(7)/4 | | | |
| 7/float(4) | | | |

| | | | |
|-------------|--|--|--|
| float (7/4) | | | |
| type(4) | | | |
| type(7/4.0) | | | |

3.3 Comparisons and boolean Expressions.

| Expression | Expected Value | Calculated Value | Reason for Calculated Value |
|------------|----------------|------------------|-----------------------------|
|------------|----------------|------------------|-----------------------------|

| | | | |
|---------------------|--|--|--|
| 3<5 | | | |
| 3<5 and 5<3 | | | |
| True | | | |
| true | | | |
| True and False | | | |
| True and True | | | |
| True or False | | | |
| False or False | | | |
| not True | | | |
| not not False | | | |
| not False and True | | | |
| not (False or True) | | | |
| not (False or True) | | | |
| (5/0==1) and False | | | |
| False and (5/0==1) | | | |

Why does the last expression in the table above “work” but the one above it doesn’t?

3.4 String Expressions.

Pay close attention to spaces and to the different types of quotation mark being used. We use both '(single quote) and " (double quote). But there is a difference between double quote " and two consecutive single quotes (")

| Expression | Expected Value | Calculated Value | Reason for Value |
|---------------------------|----------------|------------------|------------------|
| 'Truth' + 'is' + 'best' | | | |
| "Truth" + "is " + "best" | | | |
| "Truth" + ('is' + "best") | | | |
| 'A double quote: "' | | | |
| "A single quote.'"' | | | |
| 'A single quote: "' | | | |
| 'A single quote: "' | | | |
| " + 'ok' | | | |
| " + '4/2' | | | |
| " + 4/2 | | | |
| " + str(4/2) | | | |

3.5 Variables and Assignment Statements.

The last part of this lab involves assignment statements. You need to know the difference between expressions, which you've been working with so far, and assignment statements. An assignment statement like:

`b = 3 < 5`

is a command to do something. In particular, this command

- (1) evaluates the expression on the right-hand side of the = (in this case, `3 < 5`), and
- (2) stores its value in the variable on the left-hand side of the =, in this case, `b`.

Because it is not an expression, Python will not actually output a result when you type it in; it will just perform the command silently.

In the table below, the first column contains *either* an expression or a command. If it is an expression, write the value. If it is an command, you should just write "None". Because some of the entries are commands, it is important that you enter the expressions or commands in exactly the order they are given.

| Statement or Expression | Expected Value | Calculated Value | Reason for Calculated Value |
|--------------------------|----------------|------------------|-----------------------------|
| <code>i = 2</code> | None | | |
| <code>i</code> | | | |
| <code>j</code> | | | |
| <code>j = 1</code> | | | |
| <code>j = j + i</code> | | | |
| <code>j</code> | | | |
| <code>i</code> | | | |
| <code>w = 'Hello'</code> | | | |
| <code>i + w</code> | | | |