

CS 1110, LAB 1: PYTHON EXPRESSIONS

Name: _____

Net-ID: _____

There is an online version of these instructions at

<http://www.cs.cornell.edu/courses/cs1110/2012fa/labs/lab1>

You may wish to use that version of the instructions.

This lab serves two purposes. First, it is designed to get you started with Python immediately. Second, gives you hands on experience with Python expressions, which we talked about on the first day of class. With that said, you are not expected to be writing any programs for this lab; for the most part you will just be using Python as a glorified calculator.

1. GETTING STARTED WITH PYTHON

If your primary computer is a laptop, bring it to the lab as it is an *excellent* opportunity to get started with Python on your machine. Follow the instructions on the Python webpage on the course website. Feel free to ask a consultant for help if you run into problems; that is why they are there.

Even if you are just planning to use Python on the ACCEL-lab computers, you still need to learn how to use the Windows “Command Prompt” before you can get started. Even though it is not officially part of this lab, you should read the tutorial on the Command Prompt on the course web page.

Once you understand how to use the Command Prompt, you can get started by typing the word `python` at the prompt. If you are working in the ACCEL-lab, you should see output that looks something like this:

```
ActivePython 2.7.2.5 (ActiveState Software Inc.) based on
Python 2.7.2 (default, Jun 24 2011, 12:22:14)
[MSC v. 1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

To use Python, type your expression after the `>>>`. Python will evaluate the expression and provide the answer. It will then give you another `>>>` prompt so that you can type in a new expression.

There are several shortcuts that can make working with the Python prompt a lot easier. If you press the *up-arrow key*, you obtain the previous expression that you typed in. Pressing up-arrow repeatedly can scroll through all the expressions that you have typed this session. And 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* moves 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.

2. LAB INSTRUCTIONS

Below is a list of expressions. For each expression, you should 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 “?”. Once you have done that, use Python to compute the same expression. Simply cut

and past the expression from the online version of these instructions into the Interactions pane, and hit the ENTER key. You should write the result in the third column.

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 these labs (see Turning In the Lab (cf. Section 2)), so you should make your best guess at what is happening; your answer will help the instructor and consultants understand how to better aid you.

As this is the first lab, it is a little shorter than some of the future labs. With that said, do not waste time! If you don't understand something, ask your lab instructor or a consultant immediately. It is very important that you understand *how* each expression is evaluated, so if an answer doesn't make sense, ask someone. The lab instructors and consultants are in the lab to help. They will look over your shoulder from time to time and give you advice.

Turning in the Lab. Labs are graded on effort, not correctness. When you are finished, you should show your written answers to this lab to your lab instructor, who will record that you did it. You do not actually need to turn in this piece of paper; it is yours to keep. If you do not finish during the lab, finish it within the next few days and show it to your lab instructor next week.

Because the lab is graded on effort, you should focus all of your efforts on *understanding* and not just getting the answers correct. The primary purpose of these labs is for you to have a hands-on session during which the consultants and instructors can help you out.

3. LAB PROBLEMS

3.1. int Expressions.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
$2 * 3$			
$2 ** 3$			
$5 + 2 * 5$			
$(5 + 2) * 5$			
$4 - 3 - 3$			
$4 - (3 - 3)$			
$-4 - -4 - -4$			
$6 / 2$			
$6 / 4$			

Expression	Expected Value	Calculated Value	Reason for Calculated Value
$7 \% 2$			
$6 \% 3$			
$-8 \% 3$			
$2 ** 2 ** 0$			
$(2 ** 2) ** 0$			

3.2. float Expressions.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
$5.0 + 2.0$			
$9.0 * 0.5$			
$9.0 ** 0.5$			
$(5 + 2.1) * 5$			
$4.0 - 3 - 3$			
$4.0 - (3 - 3)$			
$6.0 / 2$			
$6.0 / 4$			
$6.0 \% 3$			
$-6.0 \% 4$			
$6.2 \% 4$			

3.3. Casting and Typing.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
<code>float(4)</code>			
<code>int(4)</code>			
<code>int(5.3)</code>			
<code>float(int(5.3))</code>			
<code>int(-5.3)</code>			
<code>float(7) / 4</code>			
<code>7 / float(4)</code>			
<code>float(7 / 4)</code>			
<code>7 / 4.0</code>			
<code>type(4)</code>			
<code>type(4.0)</code>			
<code>type(7 / 4.0)</code>			

3.4. boolean Expressions.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
<code>3 < 5</code>			
<code>3 < 5 and 5 < 3</code>			
<code>True</code>			
<code>True and False</code>			
<code>True and True</code>			

Expression	Expected Value	Calculated Value	Reason for Calculated Value
False			
True or False			
True or True			
not True			
not False			
not not False			
not False and True			
not False or True			
not (False or True)			
True and False and True			
True or False or True			
True or (False and True)			
True or False and True			
True and (True or False)			
True and True or False			
False and (5 / 0 == 1)			
(5 / 0 == 1) and False			

For the two expressions in the table above, answer the following question: why does the last expression in the table not work but the one above it does?

3.5. String Expressions.

In the expressions below, **pay close attention to the types of quotation marks being used**. We use three different types of quotation marks: ' (single quote), " (double quote), and ` (backquote, reached by hitting shift on the `key`). In particular, we use the backquote in the last three expressions below.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
"Truth " + "is " + "best"			
"Truth" + "is" + "best"			
"Truth " + ("is " + "best")			
"Truth " + 'is ' + "best"			
' "Truth" '			
" 'Truth' "			
" "Truth" "			
4 / 2 + ""			
"" + 4 / 2			
"" + '4 / 2'			
"" + `4 / 2`			
`Truth`			
` "Truth" `			

Given the results in the table above, answer the following question: what is the difference between the three different types of quotes?

3.6. Variables and Assignment Statements.

Up until now, you have only worked with expressions. In the table below, we have included assignment statements. It is important that you know the difference between an expression and a statement. An assignment statement like

```
b = 3 < 5;
```

is a command to do something; it evaluates the expression $3 < 5$ and stores its value in variable `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.

As with the previous part of this lab, you should complete the table below. However, you must enter the expressions or statements in the table below in **exactly the order that they are given**. Otherwise, you may get different results from everyone else. If the table row is a statement, Python will not return a value, so enter "None" for the value. We have already done this for one of the entries below.

Statement or Expression	Expected Value	Calculated Value	Reason for Calculated Value
<code>i</code>			
<code>i = 2</code>	None		
<code>i</code>			
<code>j = 1</code>			
<code>j = j + 9</code>			
<code>j</code>			
<code>k = 5;</code>			
<code>j + k</code>			
<code>w = "Hello"</code>			
<code>i + w</code>			