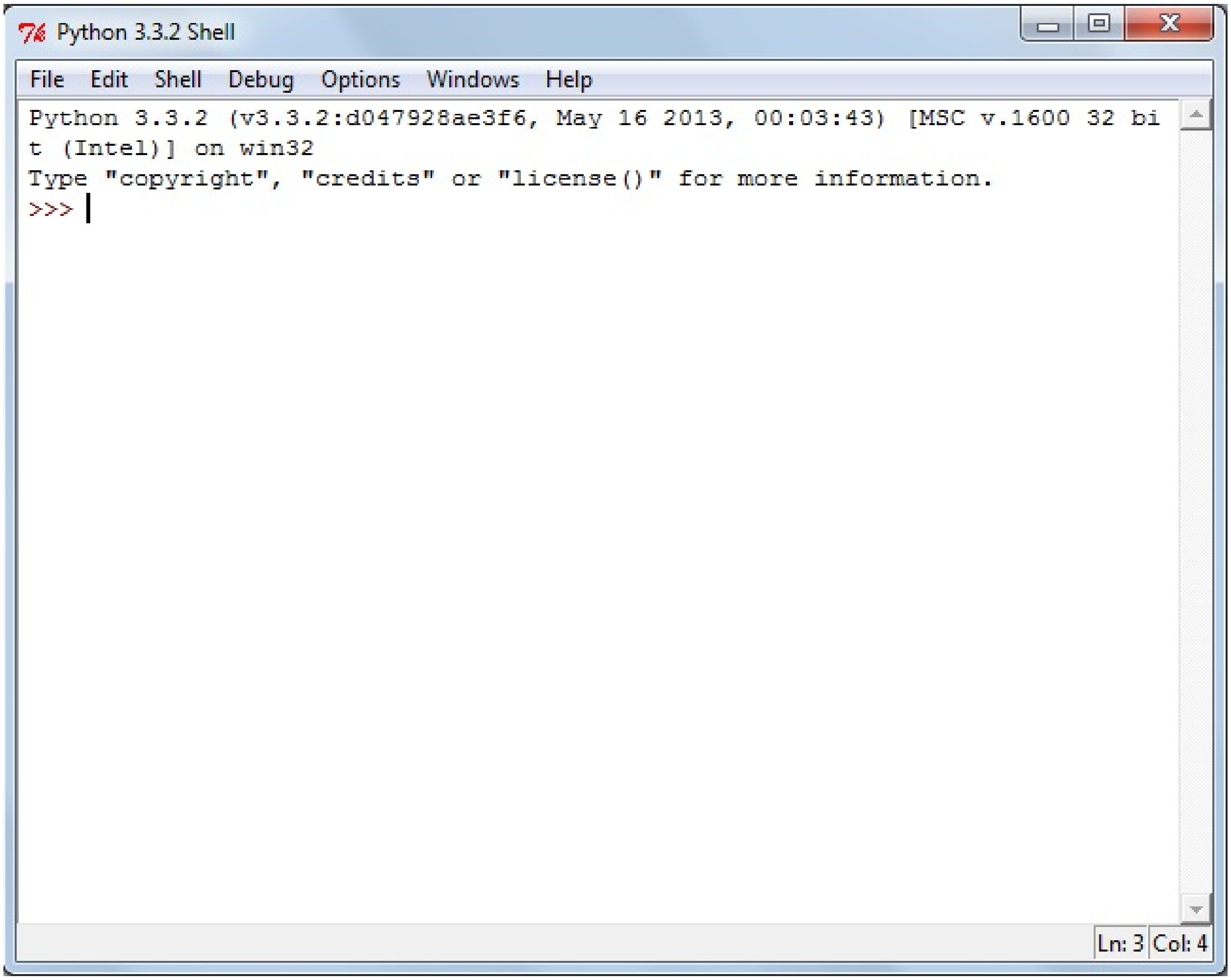
Fundamentals of Programming
Module 2 Day 2 Guided Practice

Directions:
Please print this lab and fill it in as you complete the exercises below to introduce you to programming mathematical operations, orders of precedence, and working with the Python interpreter.

Before starting this lab, make sure IDLE is open to the interpreter screen (or command prompt). This screen should open automatically when you start IDLE on your computer and should look something like this:



```
Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:03:43) [MSC v.1600 32 bi
t (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
```

The interpreter screen is great for testing small things, but it is less useful to write full programs. Today we will be using the interpreter to type in a statement, press enter, and see what Python does in response.

Up until this point, we've work with four binary operators (addition, subtraction, multiplication, and division). Python regularly uses seven binary operators. These additional operators include the **, // and % symbols. Let’s gain some practice with these new operators.

[Q1] Enter each of the following mathematical statements at the IDLE command prompt and press enter. Record what "action" the computer takes in response to your command

Arithmetic Expression	Results via the command prompt
2 ** 1	
2 ** 2	
2 ** 3	
2 ** 4	
3 ** 1	
3 ** 2	
3 ** 3	
3 ** 4	

Given what you observed in the previous step, write a short explanation of how the ** operator works in Python.

[Q2] Enter each of the following mathematical statements at the IDLE command prompt and press enter. Record what "action" the computer takes in response to your command

Arithmetic Expression	Results via the command prompt	Arithmetic Expression	Results via the command prompt	Arithmetic Expression	Results via the command prompt
8 / 4		8 // 4		8 % 4	
9 / 4		9 // 4		9 % 4	
10 / 4		10 // 4		10 % 4	
11 / 4		11 // 4		11 % 4	
12 / 4		12 // 4		12 % 4	
13 / 4		13 // 4		13 % 4	

Given what you observed in the previous step, you might have noticed that // is integer division and % is modulo (or remainder). Integer division is a lot like the mathematical floor operator and “cuts” off the decimal portion of the result. Modulo acts as remainder, and lets us know what’s left over after everything has been divided evenly.

Let’s now talk about how complex mathematical expressions are evaluated. We might remember that it isn’t just strictly from left to right. Some operations are done before others. The order of operator precedence in Python is the same as what we might have already learned for general math. From highest order of precedence to lowest, it is:

- Operations in parentheses
- Exponentiation (**)
- **Unary** negation (-) and positive (+)
- Multiplication (*), division (/), integer division (//) and remainder (%)
- **Binary** Addition (+) and subtraction (-)
- Assignment operator (=)

[Q3] Given this knowledge, **predict** what will happen when you type the following statements in the IDLE command prompt. **After** making your prediction, enter the statement at the interactions prompt and check if you were correct. If you were incorrect, determine why.

Arithmetic Expression	Predicted Results	Actual Results via the command prompt
4 * 3 + 2 * 9 / 3		
4 * 3 + 2 * (9 / 3)		
4 * (3 + 2) * 9 / 3		
20 / 4 * 6/ 2		
20 / 4 * (6 / 2)		
(20 / 4) * 6 / 2		
(20 / 4) * (6 / 2)		
(2 – 3 + (2 * (12 / 4) + 1))		
5 – (2 + 5) * 10 / 2		
4 + 2 ** 5 – 3		
-3 / 5 – 9 % 4		
-3 / + 5 – 9 % 4		
12 / 4 * - 3 + - 1		