Name_Spencer Ward_____ Date _Sept 1, 2016_____

COS 125 Lab #1 Due 11:59 PM (EoD) Weds Sept 7, 2016

SUBMIT YOUR HOMEWORK IN A SINGLE PDF FILE ON BLACKBOARD.

Type or paste your answers into this document. Save it and publish to a PDF file and submit that for this lab assignment. Your submission should include your name, the date, and the assignment number. Please give your homework file a name that includes your name and the assignment number. For example, if your name is Jane Brown, the filename could be Brown_Jane_lab1.pdf.

1. (15 points)

Install Python on your computer using the link to python.org on Blackboard to download the file appropriate for your computer and operating system. Remember to install 2.x, not 3.x. (Our TA can help you with this) Once you install Python on your computer, run the Python interpreter, and produce output that has your name and looks like the following:

```
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello, World!"
Hello, World!
>>> print 'C. Meadow says "Hello, Python!"'
C.Meadow says "Hello, Python!"
>>>
```

Include the screen output (you can usually just copy and paste between the IDLE output and your word processing document. Also try "**`Print`**" instead of "**`print`**" for one statement and paste the result.

Important note: One of the most notable differences between Python 2.x and Python 3.x is that print is a function in Python 3.x, not a statement – so you would have to enter it as

```
print("Hello, World!")
```

```
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on
win32
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello, World!"
Hello, World!
>>> print 'S. Ward says "Hello, Python!"'
S. Ward says "Hello, Python!"
>>> Print "Syntax highlighting doesn't show up in Word!"
SyntaxError: invalid syntax
>>>
```

2. (10 points) Start the Python interpreter (not IDLE) and use it as a calculator. Python syntax for math operations is almost the same as standard mathematical notation. For example, the symbols +, - and / denote addition, subtraction and division, as you would expect. The symbol for multiplication is *. Suppose you run a 10-kilometer race in 43 minutes 30 seconds. Write down a Python expression that will compute your average speed in miles per hour given that there are 1.61 kilometers in a mile.

```
>>> (10.0 / 1.61) / (43.5 / 60)
```

3. (5 points) Using the formula you came up with in the last problem, determine to the nearest whole number your speed in miles per hour.

```
9 mph
```

4. (15 points): Downey Ex.2.2 (p.13): a. Type the following statements in the Python interpreter to see what they do. What is the output? Paste or type it below

```
5
x = 5
x + 1
```

```
>>> 5
5
>>> x = 5
>>> x + 1
6
```

b. Now start IDLE, put the same statements into a script, save it as lab1-4.py and try to run it. What is the output?

```
RESTART: D:/Users/SWK/Desktop/Google Drive/Documents/2A College Work Freshman/lab1-4.py
>>>
```

c. Modify the script by transforming each expression into a print statement and then run it again. What is the output?

```
RESTART: D:/Users/SWK/Desktop/Google Drive/Documents/2A College Work Freshman/lab1-4.py
5
6
>>>
```

5. (20 points) 7: Downey Exercise 2.3 (p. 16): Using the Python interpreter, execute the following assignment statements:

```
width = 17
height = 12.0
delimiter = '.'
```

Use the Python **`type()`** function to get the type of your answer. For example, type(width) will return the type of the variable width.

If you enter the expression **`width/2`**, you will get an answer of __8__. The answer is of type __int___.

If you enter the expression **`width/2.0`**, you will get an answer of __8.5___. The answer is of type _float_.

If you enter the expression **`height/3`**, you will get an answer of __4.0___. The answer is of type _float_.

If you enter the expression **`1+2*5`**, you will get an answer of __11___. The answer is of type __int__.

If you enter the expression **`delimiter*5`**, you will get an answer of _'…..'__. The answer is of type _str_.

6. **(5 points)**. What is the difference between entering the expression **`delimiter*5`** and entering the statement **`delimiter*5`** in the Python interpreter?

Entering the expression into the interpreter tells it to to evaluate the expression and return an answer, which is formatted as **`'.....'`**, with single quotes around it, to denote it as a raw value. The statement `print delimiter*5` tells the interpreter to print the value of the expression in the command line, which is formatted as `.....`

7. (10 points) 4. Assuming the following percentage weights for grades: Exams, 35%, Homework, 20%; Projects 30% and Quizzes 15%, with grades 0-100. Assume your grades are exams = 83, homework = 91, Projects = 78 and quizzes = 82. Write expressions that will assign these values to variables and compute the final average.

```
exam = 83
homework = 91
projects = 78
quizzes = 82

examPer = .35
homeworkPer = .20
projectsPer = .30
quizzesPer = .15

finalAvg = exam * examPer + homework * homeworkPer + projects * projectsPer \
+ quizzes * quizzesPer
print finalAvg
```

Final Average: 82.95

2. (20 points) 8: Can you explain the following output? You may want to do some research on the web. Try for example "python leading zeros" You will often encounter baffling behavior by programming languages; web research skills are very useful to have.

```
>>> print 01
1
>>> print 010
8
>>> print 0100
64
>>> print 01000
512
>>> print 019
SyntaxError: invalid token>>>
```

The leading zeroes in these integers causes them to be interpreted as integers in base 8, but Python's `print` statement automatically converts any integer it prints into base 10. The reason why 019 gives an error is that the digits in base 8 only range from 0-7, so as far as Python is concerned, 019 couldn't exist as a number.