CSC 2280

Introduction to Computer Science

Florida Southern College

**Lab 1**

**Friday, January 18, 2019**

The following exercises are intended to improve your understanding of computer science, and specifically Python programming, through guided practice. Follow the instructions carefully and make sure that requested input/output looks identical to the input/output of your code.

**Notes:**

* For this lab, you may use any valid Python IDE or text editor + shell; however, it is recommended that you try PyCharm Edu as we will likely use that IDE for future labs.
* Spend some time familiarizing yourself with the IDE layout: click on various menus, look at shortcuts that may be useful, and ask questions about confusing aspects of the user interface.

1. What is the output of the following Python commands? Write your answer below each statement.   
   *Try to predict what the output will be before running the code to see if it matches your intuition!*
2. print("Hello, world!")

Hello, world!

1. print('Hello', 'world!')

Hello world!

1. print(4)

4

1. print(4.0)

4.0

1. print(4 + 4.0)

8.0

1. print(4 + 4)

8

1. print("4" + "4")

44

1. print(10\*\*0, 10\*\*1, 10\*\*2, 10\*\*3)

1, 10, 100, 1000

1. print("CSC", 2280, "is not my least favorite class!")

CSC 2280 is not my least favorite class

1. print('The answer is', round(-80 \* 2 / -4 + 2))

The answer is 42

1. print('b', 'n', 'n', '!', sep='a')

banana!

1. print(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, sep="-")

0-1-2-3-4-5-6-7-8-9

1. print(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, sep="")

0123456789

1. print("Have you",'figured out',"what sep",'does yet?', sep=' ... ')

Have you ... figured out ... what sep ... does yet?

1. print(5, "4", 3, "2", 1, sep=',', end='!')

5,4,3,2,1!

1. What does the parameter sep do in a print statement? Try several examples to familiarize yourself with the behavior of this parameter.

Fills in spaces between arguments with whatever you set the value of sep equal to

1. What does the parameter end do in a print statement? Try several examples to familiarize yourself with the behavior of this parameter.

Places whatever you make the value of end equal at the end of all the arguments.

1. What do each of the following mathematical operations do? Try out a few samples of each to verify your guesses.

|  |  |
| --- | --- |
| Operator | Operation |
| + | addition |
| - | subtraction |
| \* | multiplication |
| / | division |
| \*\* | Exponent |
| abs() | Absolute value |
| // | Whole number division |
| % | Remainder when doing whole number division |

1. What do each of the following strings (called *escape sequence characters*) do? Try out a few samples of each to verify your guesses.

|  |  |
| --- | --- |
| String | Operation |
| \\ |  |
| \' |  |
| \" |  |
| \b |  |
| \n |  |
| \r |  |
| \t |  |

I don’t understand what any of these do.

1. You can learn about a Python function by reading the help documentation. This can be accessed by running the following code:

help(function-name)

Try using the help function to describe what the following functions do?

|  |  |
| --- | --- |
| Function | What does it do? |
| print | Presents the arguments as typed |
| input | Read a string from standard input |
| int | Convert a number or string in to an integer |
| str | Creates a new string object from the given object |
| range | Produces a sequences of integers |
| list | Helps create lists, if no list is given a new one is made |

1. Write Python commands that will print the following output.  
   *Try to predict what the output will be before running the code to see if it matches your intuition!*
2. Who doesn't like contractions?

print(**'who'**, **"doesn't"**, **"like contractions?"**)

1. I am quoting my friend who once said, "This is a quote."
2. This is the first line.  
    This is the second line.

This is the third line.

1. ==========

\*\*\*\*\*\*\*\*\*\*

----------

PATTERNS!!

----------

\*\*\*\*\*\*\*\*\*\*

==========

1. 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25
2. 1 + 1 = 2
3. What is the output for the following "chunks" of code?

for i in range(5):

print(i \* i)

for d in [3,1,4,1,5]:

print(d, end=" ")

for i in range(4):

print("Hello")

for i in range(5):

print(i, 2\*\*i, sep='-->')