## Practice Assignment No. 2

**DSCI 15310 – Computational Thinking and Programming**

**September 7, 2017**

**Due Date: September 14, 2017, no later than 11:59 pm**

**My name is .**

Below is a table of two columns. The first column provides you with the information that you should type into the window of your IDLE. After you input the given instructions, put your answer into the corresponding cell in the right-hand column.

You should be observing and thinking about the results you get. You should be asking yourself, “Why did this happen?” If you have questions or don’t understand something, ask me for assistance by sending me an e-mail. You may also bring up any issues by using the discussion board.

|  |  |
| --- | --- |
| **Use the Interactive Development Environment (IDLE) to input the following information** | **What is your result?** |
|  |  |
| >>> a, b, c = range(4) | Traceback (most recent call last):  File "<pyshell#0>", line 1, in <module>  a, b, c = range(4)  ValueError: too many values to unpack (expected 3) |
| >>> number = 5  >>> w, x, y, z = range(20 / number)  >>> x | File "<pyshell#3>", line 1, in <module>  x  NameError: name 'x' is not defined |
| >>> i, j, k, l, m = range(number)  >>> l | 3 |
| >>> print("m") | m |
| >>> "m" | ‘m’ |
| >>> print ("m," "n") | m,n |
| >>> print ("m", "n") | m n |
| >>> print (SPQR) | Traceback (most recent call last):  File "<pyshell#10>", line 1, in <module>  print (SPQR)  NameError: name 'SPQR' is not defined |
| >>> print ("S" "P" "Q" "R") | SPQR |
| >>> print ("SPQR") | SPQR |
| >>> print ("S" + "P" + "Q" + "R") | SPQR |
| >>> S = "SENATUS"  >>> P = "POPULUS"  >>> Q = "QUE"  >>> R = "ROMANUS"  >>> print (S P Q R) | SyntaxError: invalid syntax |
| >>> print (S, P, Q, R) | SENATUS POPULUS QUE ROMANUS |
| >>> print (S, P + Q, R) | SENATUS POPULUSQUE ROMANUS |
| >>> x = [x1, x2, x3] | Traceback (most recent call last):  File "<pyshell#21>", line 1, in <module>  x = [x1, x2, x3]  NameError: name 'x1' is not defined |
| >>> x = [2, 4, 6]  >>> xx = x  >>> print (x, xx) | [2, 4, 6] [2, 4, 6] |
| >>> x[1] = "four"  >>> print (x) | [2, 'four', 6] |
| >>> print (xx) | [2, 'four', 6] |
| >>> print (id(x)) | 4385127752 |
| >>> print (id(xx)) | 4385127752 |
| >>> xx[2] = "six"  >>> xx[0] = "two"  >>> print (xx) | ['two', 'four', 'six'] |
| >>> print (id(x)) | 4385127752 |
| >>> print (id(xx)) | 4385127752 |
| >>> print (x) | ['two', 'four', 'six'] |
| >>> print (x, xx) | ['two', 'four', 'six'] ['two', 'four', 'six'] |
| >>> x = [1, 2, 3]  >>> print (x, xx) | 1, 2, 3] ['two', 'four', 'six'] |
| >>> print (id(x)) | 4314486728 |
| >>> print (id(xx)) | 4385127752 |
| >>> print (id(x), id(xx)) | 4314486728 4385127752 |
| >>> print (id(x) id(xx)) | SyntaxError: invalid syntax |
| >>> a = 1  >>> b = 2.0  >>> c = "iCu"  >>> d = ["a", "poor", "grade"]  >>> e = ["a", "poor", "grade"]  >>> print (id(a)) | 4297636896 |
| >>> print (id(b)) | 4300282496 |
| >>> print (id(c)) | 4385271176 |
| >>> print (id(d)) | 4314486216 |
| >>> print (id(e)) | 4385127688 |
| >>> print (id(a), id(b), id(c), \  ... id(d), id(e))  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | 4297636896 4300282496 4385271176 4314486216 4385127688 |
| >>> print (id(a), "," id(b), "," id(c), \  ... "," id(d), "," id(e))  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues on the next line. | SyntaxError: invalid syntax |
| >>> print (id(a), ",", id(b), ",", id(c), \ ... ",", id(d), ",", id(e))  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | 4297636896 , 4300282496 , 4385271176 , 4314486216 , 4385127688 |
| >>> type(a) | <class 'int'> |
| >>> type(b) | <class 'float'> |
| >>> type(c) | <class 'str'> |
| >>> Type(D) | Traceback (most recent call last):  File "<pyshell#60>", line 1, in <module>  Type(D)  NameError: name 'Type' is not defined |
| >>> type(e) | <class 'list'> |
| >>> type(a), type(b), type(c),\  ... type(d), type(e)  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | <class 'int'>, <class 'float'>, <class 'str'>, <class 'list'>, <class 'list'>) |
| >>> print (type(a)) | <class 'int'> |
| >>> type(a) == type(b) | False |
| >>> type(b) == type(c) | False |
| >>> type(c) == type(d) | False |
| >>> type(d) == type(e) | True |
| >>> type(a) != type(b) | True |
| >>> type(b) != type(c) | True |
| >>> type(c) != type(d) | True |
| >>> type(d) != type(e) | False |
| >>> type(a) == type(b) == type(c) \  ... == type(d)  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | False |
| >>> type(a) != type(b) != type(c) \  ... != type(d)  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | True |
| >>> type(a) != type(b) != type(c) \  ... != type(d) = type(e)  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | True |
| >>> type(a) != type(b) != type(c)\  ... != type(d) == type(e)  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | SyntaxError: can't assign to comparison |
| >>> type(2) == type(3) | True |
| >>> type(4.55) != type(55) | True |
| >>> type(4.55) == type(55) | False |
| >>> type("4.55") == type('55') | True |
| >>> type("Maximus") == \  ... type("Decimus") == \  ... type("Meridius")  N.B. Do not type the characters ‘\’ and the ellipsis. They just mean that the line continues onto the next line. | True |
| >>> not 0 | True |
| >>> not 1 | False |
| >>> not 555 | False |
| >>> not -555 | False |
| >>> "f" and "g" | g |
| >>> "h" and "" | '' |
| >>> 0 and "i" | 0 |
| >>> 1 and 0 | 0 |
| >>> 1 and "i" | ‘i’ |
| >>> "j" and "k" and [] and "m" | [] |
| >>> "n" or "p" | ‘n’ |
| >>> "q" or "" | ‘q’ |
| >>> 0 or "r" | ‘r’ |
| >>> 0 or 1 | 1 |
| >>> 1 or 0 | 1 |
| >>> "s" or "" or "t" or [] | ‘s’ |
| >>> 0 or "" or () or [] | [] |
| >>> 1 and "u" or "v" | ‘u’ |
| >>> "w" and "" or "x" | ‘x’ |
| >>> e = 2  >>> f = 20  >>> e == f | False |
| >>> e != f | True |
| >>> e < 3, f < 21 | (True, True) |
| >>> e < 10 < f | True |
| >>> e > 1, f == 20, 100 > f | (True, True, True) |
| >>> e <= 1, f >= 22, e == 2 | (False, False, True) |
| >>> e <= 20 == f | True |
| >>> (e >= 2) and (f == 19) | False |
| >>> (e >= 2) and (f == 20) | True |
| >>> (e != 5) or (f <= 20) | True |
| >>> (e != 5) or (f != 20) | True |
| >>> (e!= 2) or (f != 20) | False |
| >>> y = z = "Python"  >>> y is z | True |
| >>> listOne = [1, 2, 3]  >>> listTwo = [1, 2, 3]  >>> listOne is listTwo | False |
| >>> id(listOne) | 4314486920 |
| >>> id(listTwo) | 4385127688 |