University of Windsor
School of Computer Science
https://cs.uwindsor.ca

Master of Applied Computing
COMP-8347
Internet Applications and Distributed Systems

Dr. Adel Abusitta – adel.abusitta@uwindsor.ca

LAB 1 – Introduction to Python

NOTE: Use Python's **IDLE** interactive tool. Write your answer beside each command in this sheet in **bold**.

Part 1 - Lists in Python: Given the following two lists:

L1 = ['university', 'blue', 29, (4, 99, 186), [(6,'w'), 19, (5,'z')], 'DATA', 'N9E', 21, 'Delta']

L2 = ['openai', 'law', 98, 8.00, 'lion', 'extranet']

1.1 - Work with list indexing and slicing:

Indicate the results if you type the following commands in IDLE:

a) L1[2][1] => **TypeError: Int object is not subscriptable**

b) L1[3][0] => **4**

c) L1[4][2][1] => **'z'**

d) len(L1) => **9**

e) L1[14] => **IndexError: list index out of range**

f) L1[-4:-1] => **['DATA', 'N9E', 21]**

g) L1[2:14] => **[29, (4, 99, 186), [(6, 'w'), 19, (5, 'z')], 'DATA', 'N9E', 21, 'Delta']**

h) L2+L1 => **['openai', 'law', 98, 8.0, 'lion', 'extranet', 'university', 'blue', 29, (4, 99, 186), [(6, 'w'), 19, (5, 'z')], 'DATA', 'N9E', 21, 'Delta']**

i) L2*2 => **['openai', 'law', 98, 8.0, 'lion', 'extranet', 'openai', 'law', 98, 8.0, 'lion', 'extranet']**

j) L1[4][1] = 4 => **['university', 'blue', 29, (4, 99, 186), [(6, 'w'), 4, (5, 'z')], 'DATA', 'N9E', 21, 'Delta']**

k) del L2[-3] => **['openai', 'law', 98, 'lion', 'extranet']**

1.2 - Work with list methods and data types:

Type python commands to do the following:

a) append the string 'ublike' to L1 => **L1.append('ublike')**

b) remove the last element of L2 => **L2.pop()**

c) insert the item 4.8 at index 3 in L1 => **L1.insert(3,4.8)**

d) add the integers in the list [44, 50] at the end of L2 => **L2.extend([44,50])**

Part 2 - Strings in Python: Given the following two strings:

s1 = "One should note that IEEE Transactions are extremely selected"

s2 = "There are two areas in cloud computing: performance and security"

Work with string indexing, slicing, striding, assignment, concatenation: Indicate the results if you type the followingat the Python prompt in IDLE interactive mode. Indicate the type of error if the command is wrong:

a) s1[:9] => **'One shoul'**

b) s2[-1:-4] => **‘’ # empty output as the start is greater than the end index**

c) s2[-2:] => **‘ty’**

d) s2[0:15:2] => **'Teeaetoa'**

e) s1+" "+s2 => **'One should note that IEEE Transactions are extremely selected There are two areas in cloud computing: performance and security'**

Work with string methods: Use **str** methods to do the following and indicate the corresponding results:

a) Check if the string s2 ends with the word 'security' => **s2.endswith(‘security’)**

=> **True**

b) Return a list of words from s2 => **s2.split(' ')**

=> **['There', 'are', 'two', 'areas', 'in', 'cloud', 'computing:', 'performance', 'and', 'security']**

c) Convert s1 and s2 to all uppercase letters => **s1.upper(), s2.upper()**

=> ONE SHOULD NOTE THAT IEEE TRANSACTIONS ARE EXTREMELY SELECTED
=> THERE ARE TWO AREAS IN CLOUD COMPUTING: PERFORMANCE AND SECURITY

d) Replace the string 'data' of s2 with empty string => **s2.replace('data', “”)**

=> There are two areas in cloud computing: performance and security

e) Count the number of times ‘E’ occurs in s1=> **s1.count(‘E’)**

=> **3**

1) **Part 3- Dictionary in Python:** Define the following *dicts*:

#dictionary literals

d1={"name": "Alex", "age": 35, (4, 'f'):['x', 'y', 'z'], 6: "Canada", 44: 99, 19:555}

#dictionary using sequences

d2 = dict([("name","Nancy"), ('age', 44), ((3,4), ['a', 'b', 'c']), (0, 'black'), (33, 67)])

#dictionary using keywords

d3 = dict(id=777, name='Michel', siblings=['Fung', 'Martin', 'Richard'])

Work with dict methods: Type the following commands at the Python prompt in IDLE interactive mode and indicate the result of each command:

a) d1.keys() => **dict_keys(['name', 'age', (4, 'f'), 6, 44, 19])**

b) d2.values() => **dict_values(['Nancy', 44, ['a', 'b', 'c'], 'black', 67])**

c) d3.get('id') => **777**

d) d2.get('age') => **44**

e) d3.get('age') => **‘’ #which is actually None if it’s printed**

f) d3.get('name', 'Tim') => **'Michel'**

g) d2.items() => **dict_items([('name', 'Nancy'), ('age', 44), ((3, 4), ['a', 'b', 'c']), (0, 'black'), (33, 67)])**

h) d3['siblings'] => **['Fung', 'Martin', 'Richard']**

i) d2['siblings'] => **KeyError, 'siblings' not present**

j) d2.update(d3) => **d2 updated with d3 elements**

k) d2['siblings']*** => ['Fung', 'Martin', 'Richard']**

l) d2['name']***=> 'Michel'**

m) d1 == d2 => **false**

n) len(d2) => **7**

o) for key in d1.keys():

print(key)

- ⇨ **name**
- ⇨ **age**
- ⇨ **(4, 'f')**
- ⇨ **6**
- ⇨ **44**
- ⇨ **19**

p) for key in d2.keys():

print(d2[key]) *****

- ⇨ **Michel**
- ⇨ **44**
- ⇨ **['a', 'b', 'c']**
- ⇨ **black**
- ⇨ **67**
- ⇨ **777**
- ⇨ **['Fung', 'Martin', 'Richard']**