EECS 1210 Introduction to Programming in Python

Fall Semester 2022

Prof. Pai H. Chou

Self-Check 5

Due Date: --

Answer the following questions to check your understanding of your material. Expect the same kind of questions to show up on your tests.

## 1. Definitions and Short Answers

1. What is the data type of (1, 2, 3)?
2. If s = 'ABCDE', what is the value of
   * s[0]
   * s[1]
   * s[-1]
   * s[1:4]
   * s[-5:-2]
   * s[:2]
   * s[-3:]
   * s[:]
   * s[0:0]
   * s[1:4:2]
   * s[-1:0:-1]
3. If S = ['h', 'e', 'l', 'l', 'o'], what is the value of S after executing the statement S[1:2] = ['a']?
4. If T = ('h', 'e', 'l', 'l', 'o'), which of the following is allowed or not allowed and why?
   * T[3] = 'z'
   * T = ('w', 'o', 'r', 'l', 'd')
   * T = T[2:-1]
5. What is the value of
   * list('apple')
   * tuple('apple')
   * set('apple')
6. What is the value of
   * str(['a', 'p', 'p', 'l', 'e'])
   * str(('a', 'p', 'p', 'l', 'e'))
   * str({'a', 'p', 'p', 'l', 'e'})
7. What is the value of
   * list(('a', 'p', 'p', 'l', 'e'))
   * tuple(['a', 'p', 'p', 'l', 'e'])
   * set(['a', 'p', 'p', 'l', 'e'])
8. What is the result of
   * 'Apple' < 'apple'
   * 'Apple' <= 'apple'
   * 'Apple' == 'apple'
   * 'Apple' >= 'apple'
   * 'Apple' > 'apple'
   * 'Apple' != 'apple'
9. What is the result of
   * 'Apple' < 'adventure'
   * 'apple' < 'adventure'
   * 'apple' < 'Adventure'
   * 'apple' < 'bee'
   * 'apple' < 'Bee'
   * 'Apple' < 'bee'
   * 'Apple' < 'Bee'
10. What is the result of
    * ('apple', 0) < ('apple', 2)
    * ('apple', 0, 3) < ('apple', 1)
    * ['apple', 2, 2] < ['apple', 2, 1, 5]
    * ['apple', 3] < ['oranges', 0]
11. What is the result of
    * 's' in 'school'
    * 'hoo' in 'school'
    * 'S' in 'school'
    * 'ol' in 'school'
    * 'k' not in 'school'
    * 's' not in 'School'
12. What is the result of
    * 's' in ['s', 'c', 'h', 'o', 'o', 'l']
    * ['s'] in ['s', 'c', 'h', 'o', 'o', 'l']
    * ['s'] in [['s'], ['c'], ['h'], ['o'], ['o'], ['l']]
    * 'hoo' in ['s', 'c', 'h', 'o', 'o', 'l']
    * ['h', 'o', 'o'] in ['s', 'c', 'h', 'o', 'o', 'l']
    * ('h', 'o', 'o') in ['s', 'c', ('h', 'o', 'o'), 'l']
    * ('h', 'o', 'o') not in ('s', 'c', ('h', 'o', 'o'), 'l')
    * 'ol' in ['s', 'c', 'h', 'o', 'ol']
    * 's' in ['S', 'c', 'h', 'o', 'o', 'l']
13. What is the result of
    * 'sch' + 'ool'
    * [1, 2, 3] + [4, 5, 6]
    * (1, 2, 3) + (4, 5, 6)
14. What is the result of
    * 'sch' + 'o' \* 10 + 'l'
    * 'do' \* 5
    * ['s'] + ['o'] \* 5 + ['l']
15. How do you express a tuple literal of a single element? For example, how do you write a tuple literal that has the same value as tuple([1])?
16. Suppose you have x = 1, 2, 3  
    What is the value of type(x)?
17. Suppose you have L = ['f', 'r', 'o', 'g']  
    What is the new value of L after executing each of the following statements in order?
    * L.append('s')
    * L.extend(['p', 'o', 'n', 'd'])
    * L.insert(4, ' ')
    * L.reverse()
    * L.sort()
    * L.remove('o')
    * L.pop()
    * L.pop(0)
    * L.clear()
    * L.append('z')
18. If T = (1, 3, 5, 7, 9, 11), Can you call del(T[1])? why or why not? Can you call del(T)? What is the effect?
19. Suppose L = list('hello') and separately M = list('hello'). After executing  
    L.reverse()   
    M = M[::-1]
    * is L == M evaluate to True or False?
    * What is the difference between these two ways of reversing elements in a list?
20. if T = tuple('hello'), are the following statements allowed in Python? Why or why not?
    * T.reverse()
    * T = T[::-1]
21. What is a **stack** as a data structure? What is another name (4-letter initialism) for a stack? How can a stack be implemented using a list? Show how **push** and **pop** can be accomplished by calling list methods.
22. What is a **queue** as a data structure? What is another name (4-letter initialism) for a queue? How can a queue be implemented using a list? Show how enqueue and dequeue can be accomplished by calling list methods.
23. Show how a **tuple** can be used to implement
    * a stack's push and pop functionality
    * a queue's enqueue and dequeue functionality
    * Is a tuple more or less efficient than a list for implementing the stack and queue data structures? Why?
24. What do these built-in functions do?
    * max(['h', 'e', 'l', 'l', 'o'])
    * min('hello')
    * sum([2, 3, 4, 5, 6])
    * sum(range(10))
    * any(['', 'apples', 'oranges', 'banana'])
    * any([0, '', 0.0, [], ()])
    * any(['0', '', 0.0, [], ()])
    * any([0, ' ', 0.0, [], ()])
    * all(['', 'apples', 'oranges', 'banana'])
    * all([' ', 'apples', 'oranges', 'bananas'])
    * all([0, '', 0.0, [], ()])
25. What is the **non-mutation** version of the following statements? Assume L is a list
    * L.sort()
    * L.reverse()
    * L.extend([1, 2, 3])
    * del(L[1])
    * L.pop()
26. How do you use **list comprehension** to create a list with values
    * ['\*', '\*\*', '\*\*\*', '\*\*\*\*', '\*\*\*\*\*']
    * [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096]
27. How do you use **two-level list comprehension** to create a multiplication table in the following format: [(1, 1, 1), (1, 2, 2), (1, 3, 3), … (1, 9, 9), (2, 1, 2), (2, 2, 4), (2, 3, 6), (2, 4, 8), … (2, 9, 18), (3, 1, 3), (3, 2, 6), (3, 3, 9), … (3, 9, 27), (4, 1, 4) … (4, 9, 36), (5, 1, 5), … (9, 9, 81)]
28. How do you use list comprehension with filter to generate the list of upper-case letters except 'A', 'E', 'I', 'O', 'U'?
29. After executing the following sequence of statements:  
    x = 3  
    y = x  
    x = 4  
    what is the value of y?
30. After executing the following sequence of statements  
    x = [1, 2, 3]  
    y = x  
    x = [4, 5, 6]  
    what is the value of y?
31. After executing the following sequence of statements  
    x = [1, 2, 3]  
    **y = x**  
    x[1] = 4  
    what is the value of y?
32. After executing the following sequence of statements  
    x = [1, 2, 3]  
    y = **x[:]**  
    x[1] = 4  
    what is the value of y?
33. After executing the following sequence of statements  
    x = [1, 2, 3]  
    y = x  
    **y[:] =** [4, 5, 6]  
    what is the value of x?
34. After executing the following sequence of statements  
    z = ['a', 'b']  
    **x = [1, z, 3]**  
    z.append('c')  
    what is the value of x?
35. After executing the following sequence of statements  
    z = ['a', 'b']  
    x = [1, z, 3]  
    **y = x**  
    z.append('c')  
    what is the value of y?
36. After executing the following sequence of statements  
    z = ['a', 'b']  
    x = [1, z, 3]  
    **y = x[:]**  
    z.append('c')  
    what is the value of y?
37. After executing the following sequence of statements  
    z = ['a', 'b']  
    x = [1, z, 3]  
    y = **x[:]**  
    **x[0] =** 4  
    z.append('c')  
    what is the value of y?
38. After executing the following sequence of statements  
    import copy  
    z = ['a', 'b']  
    x = [1, z, 3]  
    y = **copy.copy**(x)  
    x[0] = 4  
    z.append('c')  
    what is the value of y?
39. After executing the following sequence of statements  
    import copy  
    z = ['a', 'b']  
    x = [1, z, 3]  
    y = **copy.deepcopy**(x)  
    x[0] = 4  
    z.append('c')  
    what is the value of y?
40. What is the **type** of {}?
41. What is the expression for an **empty set**?
42. Which of the following can or cannot be a **member of a set**? Why?
    * 'hello'
    * 23
    * 44.27
    * 5e-3
    * 2+4j
    * ['Mary', 'had', 'a', 'little', 'lamb']
    * ('Mary', 'had', 'a', 'little', 'lamb')
    * {'Mary', 'had', 'a', 'little', 'lamb'}
    * {'Sun': 0, 'Mon': 1, 'Tue': 2, 'Wed': 3}
    * True
    * False
    * ()
    * []
    * {}
43. What is the value of len(set('hello'))?
44. What is the value of each of the following expressions?
    * {1, 2} - {2, 3}
    * {1, 2} | {2, 3}
    * {1, 2} & {2, 3}
    * {1, 2} ^ {2, 3}
45. What is the result of the following comparisons?
    * {1, 2, 3} > {2, 3}
    * {1, 2, 3} < {1, 2, 4}
    * {1, 2, 2, 3} == {1, 2, 3}
    * {1, 2, 4} != {4, 2, 1}
46. Assume S = {1, 2, 3}, what is the difference between  
    S = S | {3, 4} and S |= {3, 4}?
47. Assume D = {'Sun': 0, 'Mon': 1, 'Tue': 2, 'Wed': 3}
    * What is the value of D['Mon']?
    * What is the value of D after D['Thu'] = 4?
    * Continuing with the previous statement, what is the value of D after D['Sun'] = 7?
    * What happens if you attempt print(D['Fri'])?
48. Assume D = {'Sun': 0, 'Mon': 1, 'Tue': 2, 'Wed': 3}
    * What is the value of D.keys()
    * What is the value of D.values()
    * What is the value of D.items()
49. Assuming D = {}, which of the following is legal or not legal in Python? If not legal, why not?
    * D[()] = 10
    * D[''] = {}
    * D[0] = ''
    * D[{}] = ()
    * D[[]] = set()
    * D[:] = range(10)
    * D[-1] = [-1]
    * D[(())] = [{}]
50. How do you use dictionary comprehension to create a reverse mapping? For example, suppose D = {'Sun': 0, 'Mon': 1, 'Tue': 2, 'Wed': 3, 'Thu': 4, 'Fri': 5, 'Sat': 6}, create its reverse mapping whose value should be {0: 'Sun', 1: 'Mon', 2: 'Tue', 3: 'Wed', 4: 'Thu', 5: 'Fri', 6: 'Sat'}?

## 2. Programming Exercises

1. Write a program that prompts the user a sentence and a list of prohibited characters. The program needs to report a list which includes all characters in the sentence except those appear in the prohibited list. For example, (blue text = typed input, green highlight = program printout)  
   $ python3 Week5\_q1.py  
   Enter a sentence: HelloPython  
   Enter prohibited list: ['l','P','h']  
   ['H', 'e', 'o', 'y', 't', 'o', 'n']

Note : Sentence is case sensitive.

1. Assume you are a server in a restaurant and you are taking the orders of a table of 4 people. They first ask you for the menu. After you give them the menu (including the meals and the prices), they take turn to order. Finally, you have to summary their orders and tell them the total bill amount.  
   Write a program to simulate the above process.  
   For example: (pink highlight = typed input, black text = program output, blue text = program output dependent to input)  
     
   $ python3 week5\_q2.py  
   customers: What are you serving today?  
   server: We have beef noodles ($100), dumplings ($60), rice ($10), vegetable ($40), tea ($30), juice ($20)  
   customer1: beef noodles, tea  
   customer2: dumplings, rice, vegetable, tea  
   customer3: rice, vegetable, tea  
   customer4: dumplings, juice  
   server: Okay, you ordered beef noodles \* 1, dumplings \* 2, rice \* 2, vegetable \* 2, tea \* 3, juice \* 1. $430 in total.