Python Data Structures & Comprehension Assignments -EURON

Lists: Creation, Methods, Slicing, Nesting

1. Create a list of 10 integers and print the first 5 elements.
2. Write a program to append and remove elements from a list.
3. Demonstrate list slicing by reversing a list using slicing.
4. Create a nested list and access a specific inner element.
5. Find the maximum and minimum value from a list without using max/min.
6. Flatten a nested list using loops.
7. Sort a list in descending order.
8. Count the occurrences of each element in a list.
9. Write a program to rotate a list left by 2 positions.
10. Merge two lists and remove duplicates.
11. Generate a list of even numbers using a loop.
12. Write a function to check if a list is a palindrome.

Deep vs Shallow Copy

13. Demonstrate shallow copy using the copy() method.
14. Create a deep copy of a nested list and prove the difference with an example.
15. Show the effect of modifying a shallow copied list on the original.
16. Use the deepcopy() method from the copy module and modify nested structure.
17. Explain deep vs shallow copy with diagrams and code.
18. Write a program to clone a list containing dictionaries.
19. Experiment with shallow copy of list of lists and explain the output.
20. Compare id() of nested elements in original and copied list.

Tuples: Packing, Unpacking, Single-element, namedtuple

21. Create a tuple and unpack it into three variables.
22. Create a tuple with one element and verify its type.
23. Swap two variables using tuple unpacking.
24. Write a function that returns a tuple of two values.
25. Use namedtuple to represent a 2D point and print coordinates.
26. Create a list of namedtuples representing students.
27. Convert a tuple to a list and back to tuple.
28. Demonstrate nested tuple unpacking.
29. Print the memory size of a tuple vs list of same elements.
30. Access elements from a deeply nested tuple.
31. Use _asdict() method on a namedtuple and display keys.

32. Find and print the index of an element in a tuple.

Sets: Properties, Methods, Set Operations, frozenset

33. Create a set from a list with duplicate values.
34. Perform union and intersection on two sets.
35. Add and remove elements using add(), discard() methods.
36. Convert a list to a frozenset and try to modify it.
37. Write a program to find unique words in a string using sets.
38. Demonstrate the use of isdisjoint(), issubset(), issuperset().
39. Create a set of squares of numbers from 1 to 10.
40. Generate a set from a string that contains only vowels.
41. Perform symmetric difference on two sets.
42. Check if two sets have at least one common element.
43. Write a function to remove duplicates using set.
44. Store unique lengths of words from a sentence in a set.

Dictionaries: Methods, Nested, View vs Copy

45. Create a dictionary and print all keys and values.
46. Write a program to merge two dictionaries.
47. Create a nested dictionary representing a student record.
48. Access and update values in a nested dictionary.
49. Use setdefault() method and explain its behavior.
50. Demonstrate difference between dict view and copy using id().
51. Count frequency of characters in a string using dictionary.
52. Invert a dictionary (keys become values and vice versa).
53. Create a dictionary from two lists using zip().
54. Write a function that takes a dictionary and returns sorted keys.
55. Clone a dictionary using copy() and modify original.
56. Delete a key-value pair safely using pop().
57. Use fromkeys() to initialize dictionary with same values.

List, Set, Dict Comprehensions

58. Use list comprehension to generate squares of 1 to 10.
59. Filter even numbers using list comprehension.
60. Create a set of unique characters from a string using set comprehension.
61. Build a dictionary of number:square for 1 to 5 using dict comprehension.
62. Flatten a 2D list using list comprehension.

63. Replace negative numbers with 0 using list comprehension.
64. Create a dictionary from a list with element:count pairs.
65. Use comprehension to get words with length > 3 from sentence.
66. Build a reverse lookup dictionary from existing key:value dict.
67. Create a set of lengths of words using set comprehension.
68. Filter dictionary to only keep items with even values.
69. Generate nested dictionary using comprehension.
70. Write comprehension to find vowels in a word and store in a list.