Date - 15/05/2024 Python Topics

Topics to cover:-

1. Data types
2. Strings
3. Booleans
4. Operators
5. List
6. Tuple
7. Sets
8. Dicts
9. Loops
10. Functions
11. Lambda
12. OOPS
13. Numpy
14. Pandas

Questions:-

1. What are decorators in Python, and how are they used?

2. Explain the difference between static, class, and instance methods in Python.

3. Describe the purpose and usage of generators and iterators in Python.

4. What are context managers, and how are they implemented using the `with` statement?

5. Explain the concept of metaclasses in Python and their practical applications.

6. Describe the principles of duck typing and how it is applied in Python.

7. What is the GIL (Global Interpreter Lock) in Python, and how does it affect concurrency?

8. Explain the purpose and usage of the `asyncio` module in Python for asynchronous programming.

9. Describe the concept of comprehensions in Python and provide examples of list, set, and dictionary comprehensions.

10. What are closures, and how are they created and used in Python?

11. Explain the difference between shallow and deep copy in Python, and when each is appropriate to use.

12. Describe the purpose and usage of the `\_\_init\_\_` and `\_\_del\_\_` methods in Python classes.

13. What is method resolution order (MRO) in Python, and how is it determined for classes with multiple inheritance?

14. Explain the purpose and usage of the `pickle` and `json` modules in Python for serialization and deserialization.

15. Describe the difference between multi-threading and multi-processing in Python, and when to use each.

16. What are context managers, and how are they implemented using the `with` statement?

17. Explain the purpose and usage of the `collections` module in Python, including `Counter`, `defaultdict`, and `deque`.

18. Describe the purpose and usage of the `logging` module in Python for logging and debugging.

19. What are Python descriptors, and how are they used to customize attribute access?

20. Explain the purpose and usage of the `\_\_slots\_\_` attribute in Python classes for memory optimization.

21. Explain the concept of method resolution order (MRO) in Python's multiple inheritance, including how it is calculated and its significance in resolving attribute lookups.

22. Describe the purpose and usage of the concurrent.futures module in Python for concurrent and parallel execution of tasks, including ThreadPoolExecutor and ProcessPoolExecutor.

Notes: Difference between list and tuple, between dictionary and set, Operators (Bitwise), relational operators, comparison operators.

Higher order functions, List and dictionary comprehension, class and object, Oops verification in Python, hierarchical and hybrid inheritance difference, wh decorator, generator, multiprocessing, and multithreading, which of them is possible in python, global interpreter lock.

Why generator, and real time applications, deep and shallow copy