Python Questions

Planned Theory Questions Related to Python:

Numpy Array vs. Python List:

- Similarities and differences between numpy arrays and Python lists.
- Numpy arrays are homogeneous and more efficient for numerical operations.

Primary Python Data Types:

- Overview of primary data types in Python, including int, float, complex, str, and bool.

List vs. Tuple:

- Differences between lists and tuples, including mutability and use cases.
- Tuples can be used as dictionary keys due to their immutability.

Python Data Structures:

- Enumeration of primary built-in data structures in Python, such as lists, tuples, sets, dictionaries, and strings.

Data Visualization in Python:

- Importing the 'matplotlib' library for data visualization, with syntax provided.
- Mention of 'seaborn' as an alternative.

Pandas DataFrame:

- Definition of a Pandas DataFrame as a 2-dimensional labeled data structure for data manipulation.

Numpy Array vs. Python List (Again):

- Description of numpy arrays as homogeneous grids with efficiency advantages over Python lists.

Python vs. Excel for Data Analysis:

- Justification for choosing Python or Excel for data analysis based on complexity and dataset size.

Additional Summary Points:

Object-Oriented Programming (OOP) in Python:

- Python supports OOP principles, and classes are used to create objects.
- An example class and its usage illustrate OOP concepts in Python.

- Python 3 introduces improvements such as print function, Unicode support, and syntax enhancements.

Exception Handling in Python:

- Exception handling is crucial for managing errors and unexpected issues in Python programs.
- Examples of try, except, and finally blocks showcase error handling techniques.

Lambda Functions in Python:

- Lambda functions are anonymous functions defined using the `lambda` keyword.
- They are often used for short, simple operations and can be passed as arguments to other functions.

The `map` Function in Python:

- The `map` function applies a given function to each item in an iterable (e.g., a list) and returns an iterable of the results.
- It is useful for applying a function to every element of a list without using explicit loops.

List Comprehensions:

- List comprehensions provide a concise way to create lists in Python.
- They combine the `map` and `filter` operations into a single expression for creating new lists.

Filtering Lists in Python:

- The `filter` function is used to filter elements from an iterable based on a specified condition.
- It returns an iterable containing only the elements that satisfy the condition.

Applying Functions to Lists:

- Functions like `map` and list comprehensions enable efficient and concise ways to perform operations on lists, making Python a powerful language for data manipulation and transformation.