

A. Core Python

1. Explain difference between `list`, `tuple`, and `set` with real-life use cases.
2. Write a function that returns the intersection of two lists (common elements).
3. Write a program to remove all duplicates from a list but keep the order.
4. Explain shallow copy vs deep copy with a code example.
5. Write a function that returns the second largest element in a list.
6. What is the difference between `is` and `==`? Show with a small example.
7. Write a function that flattens a nested list: `[[1, 2], [3, [4]]] → [1, 2, 3, 4]`.
8. Implement a function that counts frequency of characters in a string. Sample

String: `google.com` Expected Result :

```
{'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}
```

B. Functions & Problem-Solving

1. Write a recursive function to calculate factorial.
2. Implement Fibonacci sequence in three ways: `recursion`, `loop`, `generator`.
3. Explain the difference between `*args` and `**kwargs`. Write a function that uses both.
4. Explain difference between `map()`, `filter()`, and `reduce()`. Show small examples.
5. Implement binary search on a sorted list without using built-in functions.

C. OOP

1. What is the difference between a class variable and an instance variable? Show with code.
2. Explain the difference between `staticmethod`, `classmethod`, and `instance methods`, with proper code examples.
3. What are constructors and destructors in Python?
4. What is method overriding vs overloading? Does Python support both? Illustrate.
5. Write a class that overloads the `+` operator for adding two objects.
6. Write a class `BankAccount` with methods `deposit()`, `withdraw()`, `get_balance()`.

7. What is multiple inheritance? Is it supported in Python? If yes, demonstrate with example. Explain diamond problem of inheritance. How does Python solve it?
8. Explain encapsulation and how it is done in Python.

D. Small Real-World Problems

1. Given a list of transactions `[(id, amount)]`, calculate total spent per user ID.
2. Write a program that simulates a basic ATM machine (`deposit`, `withdraw`, `check balance`).
3. Write a function that compresses a string (`aaabb` \rightarrow `a3b2`).
4. Implement a simple Todo app using a class. Each Todo has `task`, `status`. Add methods to `add`, `mark done`, `list tasks`.
5. Write a function that returns all unique pairs in a list that sum to a target value.
6. Create a program that asks for a password until the correct one is entered (max 3 tries).
7. Write a program to generate a random password of length 8 containing letters and digits.