**Python Coding Interview Questions (Complete & Focused)**

**1. Basic Programming and Syntax**

* Write a Python program to **reverse a string**.
  + Example 1:
    - Input: s = ["h","e","l","l","o"]
    - Output: ["o","l","l","e","h"]
  + Example 2:
    - Input: s = ["H","a","n","n","a","h"]
    - Output: ["h","a","n","n","a","H"]
* Check if a given string is a **palindrome**.
  + Input: s = "A man, a plan, a canal: Panama"
  + Output: True
  + Explanation:
  + After cleaning: "amanaplanacanalpanama"
  + It's the same forward and backward.
* Find the **largest/smallest** number in a list.
  + Input: nums = [-10, -50, 0, 22, 7]
  + Output: Max: 22, Min: -50
* Write a function to **remove duplicates** from a list.
  + Input: nums = [1, 2, 2, 3, 4, 4, 5]
  + Output: [1, 2, 3, 4, 5]
* Count the **frequency of each character** in a string.
  + Input: s = "hello"
  + Output: {'h': 1, 'e': 1, 'l': 2, 'o': 1}
* Write a program to **check if a number is prime**.
  + Input: n = 5
  + Output: True
  + Explanation: 5 is only divisible by 1 and 5.
* Print the first **N Fibonacci numbers**.
  + Input: n = 5
  + Output: [0, 1, 1, 2, 3]
* Swap two variables **without using a temporary variable**.
  + Input: a = 5, b = 10
  + Output: a = 10, b = 5

**2. Data Structures (Lists, Tuples, Sets, Dictionaries)**

* Merge two dictionaries in Python.
  + Input:
  + dict1 = {"a": 1, "b": 2}
  + dict2 = {"b": 3, "c": 4}
  + Output:
  + {"a": 1, "b": 3, "c": 4}
* Find the **intersection and union** of two lists.
  + Input:
  + list1 = [1, 2, 2, 3, 4]
  + list2 = [2, 3, 5]
  + Output:
  + Intersection: [2, 3]
  + Union: [1, 2, 3, 4, 5]
* Write a Python program to **flatten a nested list**.
  + Input: nestedList = [[1,1],2,[1,1]]
  + Output: [1,1,2,1,1]
  + Input: nestedList = [1,[4,[6]]]
  + Output: [1,4,6]
* Find the **second largest number** in a list.
  + Input : nums = [3, 1, 4, 5, 2]
  + Output : 4
* Implement a **stack** using Python list.
* Count the number of **vowels in a string**.
  + **Input:**  
    s = "hello"
  + **Output:**  
    2
  + **Explanation:**  
    There are two vowels in "hello": 'e' and 'o'.
* Find all **duplicates** in a list.
  + **Example 1**
  + **Input:**  
    nums = [4,3,2,7,8,2,3,1]
  + **Output:**  
    [2, 3]
  + **Explanation:**  
    The numbers 2 and 3 appear twice in the list.
* Write a function to **sort a list of tuples** by the second element.
  + **Example 1**
  + **Input:**  
    tuples = [(1, 3), (3, 2), (2, 1)]
  + **Output:**  
    [(2, 1), (3, 2), (1, 3)]
  + **Explanation:**  
    When sorted by the second element, the order is (2, 1), (3, 2), (1, 3).

**3. String Manipulation**

* Check if two strings are **anagrams**.
  + **Input:**  
    s = "anagram", t = "nagaram"  
    **Output:**  
    true
* Find the **first non-repeating character** in a string.
  + **Input:**  
    s = "leetcode"  
    **Output:**  
    0  
    **Explanation:** The first non-repeating character is 'l', which appears at index 0.
* Write a program to **capitalize the first letter of each word** in a string.
  + **Input:**  
    s = "hello world from leetcode"
  + **Output:**  
    "Hello World From Leetcode"
* Implement a function to **compress a string** (e.g., aabcccccaaa → a2b1c5a3).
  + **Input:**  
    s = "aabcccccaaa"  
    **Output:**  
    "a2b1c5a3"
* Reverse words in a sentence.
  + **Input:**  
    s = "the sky is blue"  
    **Output:**  
    "blue is sky the"

**4. Algorithms and Logic**

* Implement **binary search**.
* Write a program for **bubble sort/selection sort/insertion sort**.
* Find the **missing number** in a list of consecutive numbers.
* Write a function to check if a number is a **perfect square**.
* Find the **common elements** between two lists.
* Find the **maximum sum subarray** (Kadane's algorithm).
* Calculate the **factorial** of a number using recursion and iteration.

**5. Functions and Recursion**

* Write a recursive function for **factorial**.
* Implement **Fibonacci** with recursion and iteration.
* Write a function to find the **greatest common divisor (GCD)**.
* Implement a function to solve the **Tower of Hanoi** problem.
* Write a program to calculate the **power of a number** using recursion.

**6. Object-Oriented Programming (OOP)**

* Implement a **class** for a Bank Account with deposit, withdraw, and balance methods.
* Explain and implement **inheritance** in Python with an example.
* Write a Python class with **class variables and instance variables**.
* Override a method in the child class.
* Write an example to explain **polymorphism**.
* Implement **encapsulation** with getter and setter methods.
* What are **magic methods**? Implement \_\_str\_\_ and \_\_repr\_\_.

**7. File Handling**

* Write a Python program to **read a file** and print its contents.
  + **Input:**
    - file\_path = "example.txt"
  + **Output:**
    - **"Hello LeetCode!\nWelcome to file handling in Python." (written to 'output.txt')**
* Write a program to **write data** to a file.
  + **Input**:
  + data = "Hello from LeetCode!"
  + **Output**:
  + Hello from LeetCode! (written to 'output.txt')
* Count the number of **lines, words, and characters** in a text file.
  + Input: file\_path = "sample.txt"
  + Assume sample.txt contains:
  + Hello from LeetCode!
  + Python is fun.
  + Output: (2, 6, 37)
* Append text to an existing file.
  + file\_4.txt : Hello from LeetCode!
  + **Input:** 
    - **data = "Python is fun."**
  + **Output:** 
    - **Hello from LeetCode!**
    - **Python is fun.**
* Read a **CSV file** and extract data.
  + **Input:** 
    - **file\_path = "data.csv"**
  + **Output:** 
    - **[ {"Name": "Alice", "Age": "30", "City": "New York"},**

**{"Name": "Bob", "Age": "25", "City": "London"},**

**{"Name": "Charlie", "Age": "35", "City": "Paris"}]**

**8. Exception Handling**

* Write a program with **try-except-finally** blocks.
  + **Problem Statement:**
    - Given two integers a and b, return the result of a / b.
    - If b is 0, return "Division by zero error".
    - Always print "Operation complete" after the function runs.
* Handle multiple exceptions.
  + **Description:**
    - You are given two inputs: a (an integer) and b (a divisor). Write a function that:
    - Returns a / b
    - If b is zero, return "Division by zero error"
    - If input is not an integer, return "Invalid input type"
    - Always print "Calculation attempt complete" at the end
* Create a **custom exception** and raise it.
  + **Description**:
    - Given an integer age, write a function to check if a person is eligible to vote.
    - A person must be **18 or older** to vote.
    - If age is negative or non-integer, raise a ValueError.
    - If age is less than 18, raise a custom exception UnderAgeException.

**9. Modules and Libraries**

* Import a module and use a function from it.
  + **Prompt**:  
    Write a function get\_square\_root(n: float) -> float that returns the square root of a given number n using the built-in Python math module.  
    If n is negative, raise a ValueError with the message "Input must be non-negative".
* Use **os** module to list files in a directory.
  + **Prompt**:  
    Write a function list\_files(path: str) -> List[str] that takes a directory path and returns a list of all file names (not folders) inside it using the os module.  
    Return the file names in lexicographic (alphabetical) order.
* Use **datetime** module to print the current date and time.
  + **Prompt**:  
    Write a function get\_current\_datetime() -> str that returns the current date and time as a string in the format:  
    "YYYY-MM-DD HH:MM:SS"  
    Use the Python datetime module.
* Explain how to install and use external libraries with pip.
  + **Prompt**:  
    You are given a text string. Use the external library textblob to write a function get\_sentiment(text: str) -> str that returns the sentiment of the input string.  
    Return one of: "positive", "negative", or "neutral" based on polarity score.

**10. Advanced Python Concepts**

* What is a **lambda function**? Write an example.
* Use **map(), filter(), reduce()** functions.
* Explain **list comprehensions** and write some examples.
* Difference between **deep copy** and **shallow copy**.
* Write a generator function and explain the use of yield.
* Explain **decorators** and write a simple decorator function.
* What are **iterators** and **iterables**?

**11. Miscellaneous**

* Write a program to **check if two lists are equal**.
* Implement a **queue** using collections.deque.
* Use regular expressions (re module) to validate an email address.
* Write a program to **find the median** of a list of numbers.
* Write a program to convert a list of integers to a **comma-separated string**.
* Explain **PEP 8** and write code adhering to PEP 8 guidelines.

**12. Unsolved code**

* Write a Python program to **flatten a nested list**.
  + Input: nestedList = [[1,1],2,[1,1]]
  + Output: [1,1,2,1,1]
  + Input: nestedList = [1,[4,[6]]]
  + Output: [1,4,6]
* Write a function to **sort a list of tuples** by the second element.
  + **Example 1**
  + **Input:**  
    tuples = [(1, 3), (3, 2), (2, 1)]
  + **Output:**  
    [(2, 1), (3, 2), (1, 3)]
  + **Explanation:**  
    When sorted by the second element, the order is (2, 1), (3, 2), (1, 3).
* Implement a function to **compress a string** (e.g., aabcccccaaa → a2b1c5a3).
  + **Input:**  
    s = "aabcccccaaa"  
    **Output:**  
    "a2b1c5a3"