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

**1. Basic Programming and Syntax**

* Write a Python program to **reverse a string**.
* Check if a given string is a **palindrome**.
* Find the **largest/smallest** number in a list.
* Write a function to **remove duplicates** from a list.
* Count the **frequency of each character** in a string.
* Write a program to **check if a number is prime**.
* Print the first **N Fibonacci numbers**.
* Swap two variables **without using a temporary variable**.

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

* Merge two dictionaries in Python.
* Find the **intersection and union** of two lists.
* Write a Python program to **flatten a nested list**.
* Find the **second largest number** in a list.
* Implement a **stack** using Python list.
* Count the number of **vowels in a string**.
* Find all **duplicates** in a list.
* Write a function to **sort a list of tuples** by the second element.

**3. String Manipulation**

* Check if two strings are **anagrams**.
* Find the **first non-repeating character** in a string.
* Write a program to **capitalize the first letter of each word** in a string.
* Implement a function to **compress a string** (e.g., aabcccccaaa → a2b1c5a3).
* Reverse words in a sentence.

**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.
* Write a program to **write data** to a file.
* Count the number of **lines, words, and characters** in a text file.
* Append text to an existing file.
* Read a **CSV file** and extract data.

**8. Exception Handling**

* Write a program with **try-except-finally** blocks.
* Handle multiple exceptions.
* Create a **custom exception** and raise it.

**9. Modules and Libraries**

* Import a module and use a function from it.
* Use **os** module to list files in a directory.
* Use **datetime** module to print the current date and time.
* Explain how to install and use external libraries with pip.

**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.