25 Python Function-Based Coding Questions

1. Write a function to check if a number is even or odd.

2. Write a function to return the factorial of a number.

3. Write a function to reverse a string.

4. Write a function to check if a string is a palindrome.

5. Write a function to find the maximum of three numbers.

6. Write a function to count vowels in a string.

7. Write a function to return the sum of digits of a number.

8. Write a function to return Fibonacci series up to n terms.

9. Write a function to find GCD of two numbers.

10. Write a function to check if a number is prime.

11. Write a function to return the square and cube of a number.

12. Write a function to check if a year is a leap year.

13. Write a function to return the largest element in a list.

14. Write a function to count how many times a character appears in a string.

15. Write a function to merge two lists.

16. Write a function that accepts a list and returns it sorted without using sort().

17. Write a function to find duplicates in a list.

18. Write a function to return a dictionary of character frequencies in a string.

19. Write a function to calculate simple interest.

20. Write a function to check whether a string is an anagram.

21. Write a function to return a list of even numbers from a list.

22. Write a function that accepts a number and returns its binary equivalent.

23. Write a recursive function to calculate factorial.

24. Write a function to find the second largest number in a list.

25. Write a function to check if a list is sorted or not.

15 Python OOP (Object-Oriented Programming) Coding Questions

1. Create a class Person with attributes name and age, and display them.

2. Create a class Rectangle to calculate area and perimeter.

3. Create a class Student with a method to display grade.

4. Create a class with a private variable and show how to access it.

5. Implement a class BankAccount with deposit and withdraw methods.

6. Write a program using constructor (__init__) to initialize object attributes.

7. Demonstrate single inheritance with Animal and Dog classes.

8. Demonstrate multilevel inheritance.

9. Create a class with @staticmethod and @classmethod.

10. Overload the + operator in a class using __add__.

11. Override a method in child class (method overriding).

12. Demonstrate use of super() in a subclass.

13. Create a class that keeps count of how many objects are created.

14. Create a class with __str__() method and print the object.

15. Implement a class Employee with get_salary() method and create a child class Manager.