- **Basic Level (50 Questions)**

  Focus: function definition, calling, parameters, return values, scope basics.

1. Write a function that prints "Hello, World!".
2. Create a function that takes a number and prints whether it is even or odd.
3. Write a function that adds two numbers and returns the result.
4. Define a function that returns the square of a number.
5. Create a function to find the maximum of two numbers.
6. Write a function that converts Celsius to Fahrenheit.
7. Make a function that returns the factorial of a number (using loop).
8. Write a function to calculate the area of a rectangle.
9. Create a function that accepts a name and prints a greeting message.
10. Write a function to check if a number is positive, negative, or zero.
11. Write a function that returns the length of a string.
12. Create a function that reverses a string.
13. Define a function to check if a number is prime.
14. Write a function to calculate the sum of first n natural numbers.
15. Write a function that multiplies all numbers in a list.
16. Make a function that prints the first 10 Fibonacci numbers.
17. Write a function to check if a string is a palindrome.
18. Create a function to find the minimum number in a list.
19. Write a function that returns the average of numbers in a list.
20. Define a function that prints a multiplication table of a given number.
21. Write a function that returns the largest of three numbers.
22. Create a function to count the number of vowels in a string.
23. Write a function that checks if a number is Armstrong or not.
24. Define a function that takes a list and returns a new list with unique elements.
25. Write a function to calculate simple interest.

26. Create a function that converts kilometers to miles.
27. Write a function that swaps two numbers.
28. Make a function that returns the sum of digits of a number.
29. Write a function to count the number of words in a string.
30. Define a function that returns the ASCII value of a character.
31. Write a function that returns the maximum element in a tuple.
32. Create a function that checks if a character is a vowel.
33. Write a function that calculates compound interest.
34. Make a function to return the square root of a number.
35. Write a function that takes two lists and returns their concatenation.
36. Define a function that removes duplicates from a list.
37. Create a function to check if a year is a leap year.
38. Write a function that checks if a string contains only digits.
39. Write a function that converts hours into minutes.
40. Make a function to return the cube of a number.
41. Write a function that checks if a number is divisible by another.
42. Define a function that returns the absolute value of a number.
43. Write a function to calculate perimeter of a circle.
44. Create a function that prints a list in reverse order.
45. Write a function that calculates the power of a number.
46. Make a function that counts uppercase letters in a string.
47. Write a function that counts lowercase letters in a string.
48. Define a function that finds the HCF of two numbers.
49. Create a function that finds the LCM of two numbers.
50. Write a function that checks if a number is perfect.

- **Intermediate Level (50 Questions)**

  Focus: default arguments, keyword arguments, recursion, variable scope, higher-order functions, lambda, *args and **kwargs.

1. Write a function with default parameters for a greeting message.
2. Create a recursive function to calculate factorial of a number.
3. Write a recursive function to print Fibonacci sequence.
4. Define a function that accepts *args and returns their sum.
5. Create a function that accepts **kwargs and prints them.
6. Write a function that uses both *args and **kwargs.
7. Make a function that checks if a string is an anagram of another.
8. Write a recursive function to find the sum of digits of a number.
9. Define a function that returns the nth Fibonacci number.
10. Create a function to flatten a nested list.
11. Write a function that uses a lambda to square numbers.
12. Write a function that uses map() to double elements of a list.
13. Create a function that uses filter() to get even numbers from a list.
14. Write a function that uses reduce() to compute product of numbers in a list.
15. Make a function to calculate gcd using recursion.
16. Create a function that implements binary search.
17. Write a recursive function to reverse a string.
18. Define a function that demonstrates local vs global variables.
19. Write a function that returns the intersection of two lists.
20. Create a function that returns the union of two sets.
21. Write a function that counts frequency of elements in a list.
22. Make a function that finds the longest word in a string.
23. Write a function that accepts another function as argument.
24. Create a decorator that logs the execution time of a function.
25. Write a function to check if a string is pangram.

26. Define a recursive function to find power of a number.

27. Write a function that returns prime numbers in a range.

28. Create a function that finds the second largest element in a list.

29. Write a function that counts consonants in a string.

30. Define a function that accepts a list of strings and returns the longest.

31. Write a function that checks if a number is a strong number.

32. Create a function that prints Pascal’s triangle up to n rows.

33. Write a function that uses recursion to calculate gcd.

34. Make a function that generates all prime numbers up to n.

35. Write a function that accepts a function and a list and applies the function.

36. Define a function that sorts a list without using built-in sort.

37. Write a function to check if two strings are rotations of each other.

38. Create a function that returns all factors of a number.

39. Write a function to implement linear search.

40. Define a function that accepts a string and counts digits, letters, spaces.

41. Write a recursive function to calculate sum of first n natural numbers.

42. Create a function that implements selection sort.

43. Write a function that checks if a number is palindrome.

44. Define a recursive function to generate all subsets of a list.

45. Write a function that calculates digital root of a number.

46. Create a function that prints all permutations of a string.

47. Write a function that accepts two functions and returns their composition.

48. Define a function that memoizes Fibonacci using dictionary.

49. Write a function that demonstrates closure in Python.

50. Create a function that returns a lambda function for power calculation.