1. Python (50 Questions)

1. **What are the key features of Python?**
2. **Explain the difference between lists and tuples in Python.**
3. **How does Python handle memory management?**
4. **What is the significance of the Global Interpreter Lock (GIL)?**
5. **What is a virtual environment, and why is it used in Python projects?**
6. **How do you create a virtual environment in Python?**
7. **Explain the concept of "Pass by Reference" vs "Pass by Value" in Python.**
8. **What are Python's built-in data types? Provide examples.**
9. **What is the difference between `range()` and `xrange()` (in Python 2.x)?**
10. **Explain list comprehension with an example.**
11. **What are *args and **kwargs used for in function definitions?**
12. **What is a lambda function? How is it different from a regular function?**
13. **How do you handle exceptions in Python?**
14. **What is the difference between `try-except` and `try-finally` blocks?**
15. **What are decorators in Python? Give an example use-case.**
16. **What are generators in Python? How do you create one?**
17. **What is the difference between `yield` and `return` in Python?**
18. **Explain the concept of iterators and iteration in Python.**
19. **How do you manage file I/O (reading and writing files) in Python?**
20. **What are modules and packages in Python?**
21. **How do you install external libraries in Python?**
22. **What is PEP 8, and why is it important?**
23. **What is the difference between Python 2 and Python 3?**
24. **Explain the use of `self` in Python class methods.**
25. **What is the difference between a class method and a static method in Python?**
26. **Explain inheritance in Python with a simple example.**
27. **What is multiple inheritance? How is it handled in Python?**
28. **What are `__init__` and `__new__` methods in Python classes?**
29. **Explain polymorphism in Python with an example.**
30. **What are `__str__` and `__repr__` methods used for?**
31. **What are shallow copy and deep copy?**
32. **Explain how to use the `with` statement for resource management.**
33. **What is slicing in Python? Provide examples.**
34. **What are *Pythonic* ways of programming? Give an example.**
35. **Explain the difference between `is` and == operators.**
36. **What are Python's built-in data structures (list, tuple, set, dict)?**
37. **What is a dictionary in Python? How do you iterate over it?**
38. **How do you sort a list in Python? Explain different approaches.**
39. **What are list methods like `append()`, `extend()`, and `insert()`?**
40. **Explain how Python's `in` keyword works with sequences.**
41. **How do you perform JSON parsing in Python?**
42. **What is the purpose of the `pip` tool?**
43. **What is the difference between `map()`, `filter()`, and `reduce()` functions?**
44. **Explain how to handle command-line arguments in Python scripts.**
45. **What is the purpose of the `if __name__ == "__main__":` statement?**

46. **How do you make a script executable from the command line in Python?**
47. **Explain the concept of monkey patching in Python.**
48. **How do you perform unit testing in Python?**
49. **What is a virtual environment and why do we use `venv` or `virtualenv`?**
50. **How do you work with regular expressions (regex) in Python?**

---

2. C Language (40 Questions)

1. **What are the basic data types in C?**
2. **Explain the difference between `printf()` and `scanf()`.**
3. **How does the `sizeof` operator work in C?**
4. **What is the difference between `++i` and `i++`?**
5. **Explain pointers in C. How do you declare and use them?**
6. **What is the difference between a pointer and an array?**
7. **How do you dynamically allocate memory in C?**
8. **What is the purpose of `malloc()`, `calloc()`, `realloc()`, and `free()`?**
9. **Explain the concept of pointer arithmetic.**
10. **What are function pointers and how can they be used?**
11. **What is a segmentation fault? Give common reasons it occurs.**
12. **Explain the structure of a C program (header files, main function, etc.).**
13. **What are storage classes in C? (auto, static, extern, register)**
14. **Describe how arrays and strings are handled in C.**
15. **Explain the difference between `struct` and `union`.**
16. **How do you pass an array to a function in C?**
17. **What is recursion? Provide a simple example in C.**
18. **What is `typedef` used for in C?**
19. **What are macros, and how do you define them in C?**
20. **Explain the difference between `#include <file>` and `#include "file"`.**
21. **How do you handle command-line arguments in C?**
22. **What is the function prototype in C?**
23. **What are header guards and why are they used?**
24. **Explain the concept of undefined behavior in C.**
25. **What is the difference between `break` and `continue` statements?**
26. **How do you implement error handling in C?**
27. **What is the difference between `exit(0)` and `exit(1)`?**
28. **Explain how to return multiple values from a function in C.**
29. **What is the purpose of the `volatile` keyword?**
30. **How do you swap two variables without using a third variable in C?**
31. **Explain the difference between `const char* p` and `char* const p`.**
32. **What is the difference between `getc()`, `getchar()`, `scanf()`, and `fgets()`?**
33. **Explain how to read and write files in C.**
34. **What is a function pointer array? Provide a use case.**
35. **Explain the difference between compilation and linking.**
36. **What is the role of a makefile in C projects?**
37. **Describe the process of passing structures to functions.**
38. **What are enumerations (`enum`) in C?**
39. **How does type casting work in C?**

40. **Explain the difference between call by value and call by reference in C.**

---

3. Data Structures & Algorithms (DSA) (40 Questions)

1. **What is an array? How is it different from a linked list?**
2. **Explain the differences between a stack and a queue.**
3. **What is a linked list? Describe its types (singly, doubly, circular).**
4. **How do you reverse a linked list?**
5. **Describe the concept of a stack and its operations (push, pop, peek).**
6. **Explain queue operations (enqueue, dequeue).**
7. **What is a priority queue? How is it implemented?**
8. **Explain the concept of hashing.**
9. **What is a collision in hashing, and how can it be handled?**
10. **Describe the difference between a binary search tree (BST) and a heap.**
11. **What is the difference between a binary tree and a BST?**
12. **How do you traverse a binary tree (inorder, preorder, postorder)?**
13. **What is a balanced binary tree? Give examples.**
14. **Explain depth-first search (DFS) vs breadth-first search (BFS).**
15. **Describe how to implement a graph using adjacency list and adjacency matrix.**
16. **What is the difference between directed and undirected graphs?**
17. **Explain topological sort. In which scenarios is it used?**
18. **What is the time complexity of common sorting algorithms (Quick, Merge, Bubble, Insertion, Selection)?**
19. **Explain merge sort and its complexity.**
20. **What is the worst-case complexity of quicksort, and how can it be improved?**
21. **Compare the advantages of merge sort over quicksort and vice versa.**
22. **Explain binary search. What is its time complexity?**
23. **How do you find the middle element of a linked list in one pass?**
24. **What is a Fibonacci series? Write a simple algorithm to generate it.**
25. **Describe the concept of dynamic programming. Give an example problem.**
26. **What is memoization? How is it different from tabulation in DP?**
27. **Explain the concept of greedy algorithms. Provide an example.**
28. **How do you detect a cycle in a linked list?**
29. **Explain the Floyd's cycle-finding algorithm.**
30. **What is a spanning tree? Differentiate between MST algorithms (Kruskal's, Prim's).**
31. **Describe the shortest path algorithms (Dijkstra, Bellman-Ford).**
32. **What is the difference between a recursion and an iteration approach?**
33. **Explain how to find the maximum subarray sum (Kadane's algorithm).**
34. **How do you determine if a given tree is a BST?**
35. **Explain the concept of backtracking with an example (e.g., N-Queens).**
36. **What are the different ways of collision resolution in hashing (chaining, open addressing)?**
37. **Explain how to implement a min-heap and max-heap.**
38. **What is the time complexity of heapify and building a heap?**
39. **How do you perform a heap sort, and what is its complexity?**
40. **Explain the difference between best case, average case, and worst case complexity.**

4. Java OOPS (40 Questions)

1. **What is OOP, and why is it important?**
2. **Explain the four pillars of OOP (Encapsulation, Inheritance, Polymorphism, Abstraction).**
3. **How do you define a class and an object in Java?**
4. **What is the difference between a constructor and a method?**
5. **Explain method overloading vs method overriding in Java.**
6. **What is inheritance? Provide an example.**
7. **What are access modifiers in Java (public, private, protected, default)?**
8. **Explain the concept of encapsulation with a code example.**
9. **What is the difference between an interface and an abstract class?**
10. **Explain how multiple inheritance is handled in Java.**
11. **What is the significance of the `this` keyword?**
12. **What is the use of the `super` keyword in inheritance?**
13. **How does Java achieve polymorphism?**
14. **Explain dynamic method dispatch with an example.**
15. **What is the difference between compile-time and run-time polymorphism?**
16. **How do you prevent a class from being extended in Java?**
17. **What is the role of the `static` keyword in Java?**
18. **Explain the difference between instance variables and class variables.**
19. **What are inner classes in Java? Why use them?**
20. **Explain the concept of packages in Java.**
21. **What is the significance of the `final` keyword (class, method, variable)?**
22. **Describe the constructor chaining in Java.**
23. **What is an abstract method? Give an example.**
24. **Explain the difference between `==` and `.equals()` in Java.**
25. **How does Java handle memory management and garbage collection?**
26. **What is the difference between Stack memory and Heap memory?**
27. **Explain the `String` pool in Java.**
28. **What is the difference between `String`, `StringBuilder`, and `StringBuffer`?**
29. **Describe how exceptions are handled in Java.**
30. **What is the difference between checked and unchecked exceptions?**
31. **What is the purpose of the `throws` keyword in a method signature?**
32. **Explain try-with-resources statement in Java.**
33. **What are generics in Java? Why are they used?**
34. **What are collections in Java? Name some common collection classes.**
35. **Explain the differences among `ArrayList`, `LinkedList`, and `Vector`.**
36. **What is the difference between `HashMap` and `Hashtable`?**
37. **How do you iterate over a collection in Java?**
38. **Explain the concept of autoboxing and unboxing.**
39. **How do you implement a custom exception in Java?**
40. **What are the main principles to write clean and maintainable OOP code in Java?**

5. HTML & CSS (40 Questions)

1. **What is the basic structure of an HTML document?**
2. **Explain the difference between block-level and inline elements.**
3. **What are semantic HTML elements? Give examples.**
4. **What is the purpose of the `<!DOCTYPE html>` declaration?**
5. **How do you include external CSS in an HTML page?**
6. **Explain the difference between `<link>` and `@import` for CSS.**
7. **What are the different types of CSS (inline, internal, external)?**
8. **Explain the CSS box model.**
9. **What is the difference between `margin` and `padding`?**
10. **What are CSS selectors? Describe different selector types.**
11. **How do you target elements by ID vs class in CSS?**
12. **What is the difference between `display: none` and `visibility: hidden`?**
13. **What is the purpose of the `float` property in CSS?**
14. **Explain how to clear floats in CSS.**
15. **What is the difference between relative, absolute, fixed, and static positioning?**
16. **How do you center a `<div>` horizontally using CSS?**
17. **What are pseudo-classes and pseudo-elements in CSS?**
18. **Explain how media queries are used for responsive design.**
19. **What is the difference between `em` and `rem` units?**
20. **How do you implement CSS resets or normalize CSS?**
21. **What are HTML5 form elements and how are they useful?**
22. **Explain the `<canvas>` element in HTML5.**
23. **What is the difference between `<section>` and `<div>`?**
24. **How do you optimize a website's performance from a CSS perspective?**
25. **Explain the concept of specificity in CSS.**
26. **What are `inline-block` elements in CSS?**
27. **How do you use Flexbox for layout?**
28. **What is CSS Grid, and how does it differ from Flexbox?**
29. **How do you use transitions and transformations in CSS?**
30. **What is the purpose of vendor prefixes (e.g., `-webkit-`, `-moz-`)?**
31. **Explain the concept of responsive design.**
32. **How do you incorporate fonts using `@font-face`?**
33. **What is the difference between `position: absolute` and `position: fixed`?**
34. **Describe how to create a dropdown menu in pure HTML/CSS.**
35. **How do you add comments in HTML and CSS?**
36. **What is the difference between `<span>` and `<div>` elements?**
37. **How do you handle browser compatibility issues in CSS?**
38. **What are `meta` tags in HTML, and why are they important?**
39. **How can you embed audio and video in HTML5?**
40. **Explain how to make images responsive in HTML/CSS.**

6. JavaScript (40 Questions)

1. **What are the different data types in JavaScript?**
2. **Explain the difference between `var`, `let`, and `const`.**
3. **What is hoisting in JavaScript?**
4. **Explain how scoping works in JavaScript.**
5. **What are closures? Provide an example.**
6. **How does JavaScript handle asynchronous operations?**
7. **What is the event loop in JavaScript?**
8. **Explain callback functions and give an example scenario.**
9. **What are promises in JavaScript?**
10. **What is `async/await`, and how does it simplify asynchronous code?**
11. **Explain the concept of `this` keyword in JavaScript.**
12. **What is prototypal inheritance, and how does it differ from classical inheritance?**
13. **Explain the difference between == and ===.**
14. **What are arrow functions, and how are they different from regular functions?**
15. **What is the difference between `null` and `undefined`?**
16. **Explain how to handle errors in JavaScript (try/catch/finally).**
17. **What is the difference between synchronous and asynchronous code?**
18. **How do you manipulate the DOM using JavaScript?**
19. **Explain event bubbling and event capturing.**
20. **What are higher-order functions in JavaScript?**
21. **Describe the difference between `map()`, `forEach()`, `filter()`, and `reduce()`.**
22. **What are template literals in ES6?**
23. **Explain how to clone an object in JavaScript.**
24. **What are modules in JavaScript (import/export)?**
25. **Explain the concept of `strict mode`.**
26. **How do you debounce or throttle a function in JavaScript?**
27. **What are the different ways to create objects in JavaScript?**
28. **What is the difference between function declaration and function expression?**
29. **Explain how JSON is used in JavaScript.**
30. **What are some common array methods in JavaScript?**
31. **How do you handle cross-browser compatibility in JavaScript?**
32. **What is the DOMContentLoaded event?**
33. **Explain the difference between `window.onload` and `document.onload`.**
34. **What are Web APIs in JavaScript? Provide examples.**
35. **What is the Fetch API, and how does it compare to XMLHttpRequest (XHR)?**
36. **Explain how localStorage and sessionStorage work.**
37. **What is the purpose of the `new` operator in JavaScript?**
38. **How do you create and dispatch custom events in JavaScript?**
39. **What is event delegation? Why is it useful?**
40. **Explain how you would debug JavaScript code.**

7. SQL & NoSQL (40 Questions)

1. **What is a relational database? Give some examples of RDBMS.**
2. **Explain the difference between SQL and MySQL.**
3. **What are the different types of SQL statements (DDL, DML, DCL, TCL)?**
4. **Explain the concept of primary key, foreign key, and unique key.**
5. **What is normalization, and why is it used?**
6. **Explain the different normal forms (1NF, 2NF, 3NF, BCNF).**
7. **What is denormalization? When would you use it?**
8. **How do you use the `JOIN` clauses (INNER, LEFT, RIGHT, FULL)?**
9. **What is the difference between `WHERE` and `HAVING` clauses?**
10. **Explain the concept of subqueries in SQL.**
11. **What is an index, and how does it improve query performance?**
12. **What are clustered and non-clustered indexes?**
13. **What is the difference between `DELETE` and `TRUNCATE`?**
14. **Explain the concept of transactions in SQL.**
15. **What are ACID properties in database transactions?**
16. **How do you implement a stored procedure in SQL?**
17. **What is a trigger? Give an example scenario.**
18. **How do you use aggregate functions (COUNT, SUM, AVG, MIN, MAX)?**
19. **Explain the `GROUP BY` clause and when to use it.**
20. **What is the difference between `UNION` and `UNION ALL`?**
21. **How do you handle NULL values in SQL?**
22. **What are constraints in SQL (NOT NULL, CHECK, DEFAULT)?**
23. **Explain the difference between a database view and a table.**
24. **What are common performance optimization techniques in SQL?**
25. **How do you backup and restore a database?**
26. **What is a NoSQL database, and how is it different from SQL databases?**
27. **Explain the main categories of NoSQL databases (Key-Value, Document, Column, Graph).**
28. **When would you choose NoSQL over SQL?**
29. **Give examples of popular NoSQL databases (MongoDB, Cassandra, Redis, Neo4j).**
30. **Explain the CAP theorem in the context of distributed databases.**
31. **How does sharding work in NoSQL databases?**
32. **What is replication, and how is it implemented in NoSQL databases?**
33. **How do you query data in MongoDB (basics of find, insert, update, delete)?**
34. **What is the difference between a collection and a document in MongoDB?**
35. **Explain eventual consistency in NoSQL databases.**
36. **What is MapReduce, and how is it used in NoSQL systems?**
37. **How do you handle schema changes in NoSQL databases?**
38. **Explain the difference between vertical scaling and horizontal scaling.**
39. **What are some challenges of using NoSQL databases?**
40. **In what scenarios is a relational SQL database still preferred over NoSQL?**
