Project:

1. Be sure about the entire project workflow.
2. Prepare one module completely (with code)
3. Make sure that you have sound theoretical knowledge on the framework which you have used in the project. (*In case of spring boot, prepare the concept of maven. Maven concept is very important.)*

Certifications:

1. If you add global certifications, get to know the basic meta data of the certification.
2. AWS: Main areas to focus on-
   a. EC2
   b. S3
   c. DATA BASES
   d. ECS
   e. LAMBDA
   f. IAM
   g. Cloud9

Prepare accordingly if it is azure or GCP.

C - PROGRAMMING CONCEPTS:

1. Dynamic Memory Allocation
2. Structures and Unions – Padding concept is important here.
3. Pointers
   a. Pointer to a pointer – (Double Pointer)
   b. Pointers and arrays
   c. Pointers and strings
   d. Pointers and structures
   e. Pointers and functions
   f. Null pointer and void pointer – difference is important.
   g. Pass by value and pass by reference.
   h. Pointer de-reference.
4. Operators
   a. Operator precedence and associativity
   b. Post – increment, pre – increment
   c. Post – decrement, pre – decrement

NOTE: INFIX, PREFIX, POSTFIX EXPRESSIONS AND CONVERSIONS

Bit Manipulation:

1. Counting number of set bits
2. Swap two number without using a third variable
3. Check if a number is a power of 2
4. Given an array, find the single non repeating element
5. Describe the applications of bitwise operators.
6. Reverse the bits of an unsigned integer
7. Given two integers, find the bitwise AND of all numbers in the range of 2 integers(inclusive).

Java:

1. Memory management concepts in java
2. Execution flow of a java program
3. Command line arguments
4. JDK, JRE, JVM – all concepts
5. Why setter and getter methods?
6. Chart of Data types with ranges
7. Static and non-static concepts
   a. Static with variables
   b. Static with methods
8. Every important point about main method
9. Primitive and Non-primitive Data types - important differences
10. Wrapper classes
11. Type Conversion – Parse Methods

OOPS

IMPLEMENTATION OF:

a. Inheritance
b. Abstraction
c. Polymorphism
d. Encapsulation

ALL IMPORTANT POINTS OF ABOVE CONCEPTS.

Explain the concept of access modifiers in java by relating with a real-life example!

Explain all the OOPs features with real life examples.

Exception Handling:

1. Types of exceptions.
2. Every minute detail about each type
3. Try, catch, and finally block combinations.
4. Use of Finally block.
5. Try, catch, and finally block combinations.
6. Different keywords in exception handling in java
7. Difference between throw and throws keyword.
8. Knowledge on customised exception.
9. Difference between final, finally and finalise keywords in java.

NOTE: Multithreading concept is very important for product-based company interviews.

Collection Framework:

1. Hierarchy of collection framework in java.
2. Make sure that the properties table is crystal clear in your mind. Helps to learn differences between them easily and quickly.
3. *Internal working of hashset.*

Coding:

String Manipulation:

a. CheckIfALLUniqueCharOrNot
b. CheckPalindrome
c. CheckIfAnagram
d. CheckIfPanagram
e. CheckIfStringIsEmpty
f. CheckOnlyForIntegers
g. CovertLowerCaseToUpperCase
h. CopyStringToOther
i. CountVowelsConsonantsSymbols
j. PrintAllDuplicateCharacters
k. ConvertFirstLetterOfWordToUpperCase
l. FindFirstNonRepeatingCharacter
m. FindFrequencyOfCharacters
n. PrintLastWordInAString
o. LongestWordInAString
p. MaxOccuringCharacter
q. PrintAllNonRepeatingCharacters
r. NumberOfWordsInString
s. PrintWordsWithFrequency
t. ProveStringsAreImmutable
u. RemoveDuplicates
v. RemoveWhiteSpacesOfAllKind
w. Reverse
x. SwapTwoStringsWithTempVariable
y. SwapTwoStringsWithoutTempVariable
z. ScoreAndGood
aa. FindLongestSubSequence
bb. FindLongestSubString

NOTE: Description of ‘z’ ->

```
 //if the number of vowels in the string is more
//than the number of consonants ---> it is a good string
//Score of the string is the sum of the digits in it
```

Theory Questions:

1. Why Strings are Immutable in java
2. Char array or String, which is used to store passwords in java
3. How memory is allocated to string objects based on the various types of declarations and initialisations in java
4. What is String constant pool? What is the other name?
5. What is a String Literal?
6. What is the difference between String and StringBuilderName some important methods in String Class
7. Explain the concept of memory management with respect to strings in

java

8. What is the advantage and disadvantage of stringbuilder over string.
9. Give some real life examples of where strings and String builder are used.
10. Is String Thread-safe in java
11. How do u compare 2 strings in java. What are the 2 basic important comparisons. Name the operators used for the comparison.
12. What is intern() method in java?
13. Name some common string manipulation methods
14. What is the difference between substring and subsequence

Array Manipulation

One Dimensional:

1. CommonElementsIn2ArraysUsingLoops
2. CommonElementsIn2ArraysUsingHashSet
3. CountSumProductOfEvenAndOdd
4. CommonElementsIn3SortedArrays
5. DeleteElement
6. AddElement
7. DuplicateElementsBruteForce
8. DuplicateElementsUsingSet
9. FirstDuplicateElementBruteForce
10. FirstDuplicateUsingLinkedHashMap
11. FirstDuplicateElementUsingHashMap
12. KadanesAlgorithm
13. KthLargestElement
14. KthSmallestElement
15. LongestConsecutiveSequence
16. MaximumAndMinimum
17. MissingElementArraySumMethod
18. MissingElementUsingXor
19. Merge2arrays
20. PairsWithGivenSum
21. RemoveDuplicates
22. ReverseArray
23. SecondLargestElement
24. UniqueElementsUsingXor
25. UniqueElementsWithSet
26. FirstDuplicateElement
27. LengthOfLargestContiguousSubArray
28. CloneTheArray

TwoDimensional:

1. AverageOfDiagonalElements
2. LargestElementIn2DArray
3. MatrixAddition
4. MatrixMultiplication
5. ReverseRows
6. ReverseColumns
7. RotateMatrixBy90degree
8. SearchElementInFullyRowSortedMatrix
9. SumOfDiagonalElements
10. SumOfLeftDiagonalElements
11. SumOfRightDiagonalElements
12. Transpose
13. FindSaddlePoint
14. MagicSquareProblem
15. SpiralTraversal
16. DiagonalTraversal

Theory Questions:

1. Explain the concept of Dynamic Memory ALLOCATION with respect to arrays.
2. List out all the ways of declaring a 2 D array – VIP Question

Searching:

1. Linear Search

a. On Integers
b. On Strings
c. Real life application on linear search

2. Binary Search
    a. Real life applications of binary search
    b. Is it true that binary search can be applied on an array only if it is sorted. – (Tricky question)
    c. Revised Formula of mid. Why should we consider this? – V.V.V.V.V.V.V.V.V.I.P – (Asked in all interviews)
    d. Problems on B.Search: Index of first and last occurrence in sorted array, count of 1's in a binary sorted array, peak element, square root of an integer, finding an element in an infinte sorted array, finding an element in a sorted and rotated array.

Two Pointer Approach:

1. Find pair in an unsorted array which sum to X
2. Find pair in a sorted array which sum to X
3. Find triplet in array which sum to X

Sorting:

1. BubbleSort
2. MergeSort
3. QuickSort
4. InsertionSort
5. HeapSort
6. SelectionSort (On Stirngs Also)
7. StringSort

Note: You should be able to write complete code for merge sort and quick sort. (Important for product-based companies)

Bubble sort, selection and insertion sort are obvious things.

Grab all the information regarding time complexities of every sorting algorithm by considering all the cases. - V.V.V.I.P

(I faced a question which goes as follows:

Write time complexities by considering various scenarios for every sorting algorithm you know and explain it with an example)

Linked List:

1. Implement Singly , Doubly, Circular , Circular Doubly Linked List
   a. Addition:
      Add at begin
      Add at end
      Add at specified Position
   b. Deletion
      Delete at
      start Delete
      at end
      Delete at specified Position
2. Remove Duplicates
3. Count the number of nodes
4. Merge two sorted Linked List
5. Detect Cycle in Linked List
6. Remove Cycle in Linked List
7. Find the Intrsection Point of two Linked List / Starting point of the cycle
8. Reverse a Linked List
9. Reverse a Linked List of K size
10. Rotate a linked list – left rotation and right rotation, how many times rotate
11. Check if linked list is palindrome
12. Find the middle point of the linked list – count method and slow-fast pointer approach
13. Alternate merge of 2 linked llist
14. Flattening a Linked List
15. Find and return address of previous node of given data
16. Find the given data in linkedlist
17. Arrange the Linked List in front-last pair
18. Find the last occurrence of given data

19. Add two numbers
20. Clone a Linked List

Theory Questions:

a. Real life applicationsof linked list – Very Important Question

b. Explanation about all the types of Linked List

c. Explain how memory allocated for nodes in a linked list

d. How does DMA differ between array and Linked List

e. Demonstrate time complexity of insertion, deletion, and search operations considering all the possible positions(start,end,middle)

f. Which algorithms are used to detect cycles in a linked list

g. How to implement stack using LL

h. How to implement queue using LL

Stack:

1. Implement your own stack by using arrays and linked list.
2. Which implementation of stack is better – array or linkedlist – why?
3. Implement stack using queue
    a. Implement push efficient stack
    b. Implement pop efficient stack
4. Real life applications of stack
5. Computer based applications of stack
6. Balanced paranthesis
7. Reverse stack using queue
8. Reverse string using stack
9. What is the need of prefix and postfix notations
10. Next greater element
11. Next smaller element
12. Previous greater element
13. Previous smaller element
14. Infix to postfix
15. Clone a stack without extra space
16. Design a stack that supports retrieving min element in O(1) time complexity
17. Can u Reverse a stack without extra space? If yes how?
18. Sort a stack using recursion
19. Reverse stack using recursion

Queue:

1. Implement queue using arrays
2. Implement queue using linked list
3. Implement queue using 1 stack.
4. Implement queue using 2 stacks.
5. Implement Circular queue using arrays Queue full condition is very important here.

6. Implement circular queue using linked list.
7. Real life examples of queue
8. Computer based examples of queue
9. Count number of islands

Recursion:

1. Applications of recursion
2. Writing base cases in recursion (factorial, nth Fibonacci number)
3. Print 1 to n and n to 1.
4. Sum of digits
5. Subsets of a set
6. Tower of Hanoi

DBMS:

1. Architecture
2. Definition of data, information, and difference between them. -> VIP question, faced in 2 product-based company interviews.
3. ER concepts.
4. Types of keys - VIP
5. Checkpoint concepts
6. Normalisation – dependency def, types of dependencies, removal of dependencies
7. Types of languages – VIP
8. Relational Algebra – VVIP for product company interviews. This can be a point where the flow of the interview gets disturbed, and control goes from the candidate to the interviewer.
9. Data abstraction and levels of data abstraction
10. Query optimisation – VIP
11. Execution flow of a SQL query
12. ACID properties – VIP
13. Foreign key and Data Integrity concept
14. Types of constraints
15. Indexing concepts
16. Clustered and non-clustered indexes
17. Transaction management
18. Types of locks
19. Explain 1NF,2NF,3NF, BCNF by taking your own example tables – VVVVIP – Faced in 1 interview
20. What is a view. Use of view
21. Trigger concepts
22. Stored Procedure concepts
23. Cursor concepts

QUERIES

1. What is the need of join concept – explain with an example – VIP – Faced in 1 product-based company interview.
2. JOIN def and types of joins!
3. Explanation of each join with examples – especially self-join – faced in 1 interview.
4. Clauses in SQL – where, having, limit,group by, order by, over , partition by , WITH
5. DISTINCT keyword important points
6. Why select is listed in DML? Faced in 1 interview.
7. Rank and Dense rank

8. Finding nth highest salary in a table – explain in 4 ways – VVVVVVVIP – Faced in 2 interviews.
9. Difference between having and where clause.
10. Operators – IN, NOT IN, BETWEEN, LIKE (2TYPES OF WILD CARD CHARACTERS OF LIKE OPERATOR), AND, OR, NOT
11. Queries based on cursors, triggers, and stored procedures – Give a glance.
12. Different types of constraints
13. Difference between UNION and UNION ALL
14. Subquery – corelated subquery.
15. Case manipulation functions in SQL
16. Datatypes in SQL
17. Difference between char(n) and varchar(n) – VVVVVIP – Asked in 3 interviews, just definition is not enough. Be sure about the ranges also.
18. What is on delete cascade.
19. What is default constraint.
20. What is alias.
21. What are aggregate functions – MIN, MAX, SUM, AVG – **Be careful when u use aggregate functions in the query. Because u need to group the non-aggregated columns based on some property.**
22. COMMIT, ROLLBACK and SAVEPOINT – example queries

Operating System:

1. Paging all concepts
2. Segmentation all concepts
3. Process scheduling all concepts – most importantly – algorithms.
4. Deadlock, deadlock prevention
5. Semaphore and its types
6. Virtual memory concept
7. Hypervisor concept
8. Different states of process
9. Thrashing concept
10. Difference between multiprogramming and multitasking
11. Demand paging def
12. Levels of RAID
13. System process and user process – terminology and related concepts is important.
14. User level thread and system level thread?
15. What is logical and physical address spaces?
16. Problems of classic synchronisation
17. Premptive and non-pre-emptive differences
18. IPC and Various IPC mechanisms
19. Starvation and aging
20. Context switching
21. Different types of kernels - monolithic and other types – imp
22. Diff btw program and process
23. Diff btw thread and process
24. What is critical section?
25. Name some synchronisation techniques!
26. What is Peterson’s approach?
27. Bankers algo
28. Bounded waiting
29. Concurrency
30. Resource allocation graph explanation

Software Engineering:

1. SDLC lifecycle. List out the phases in SDLC.
2. What is software.
3. Characteristics of software
4. Categories of software
5. What are Umbrella activities
6. Blackbox testing and white box testing
7. Agile Software Development
8. Difference between Traditional and agile software development
9. Comparison between agile model and other models
10. Difference between SRS and FRS
11. Difference between Quality Assurance and Quality Control
12. Difference between verification and validation
13. What is re-engineering?
14. What is reverse engineering?
15. Difference between Alpha testing and beta testing
16. Difference between risk and uncertainty.
17. What is a use case diagram?
18. Software development models
19. Software Project Management
    a. Waterfall
    b. Iterative waterfall
    c. Prototyping model
    d. Incremental model
    e. Spiral model
    f. RAD

20. All the important UML diagrams

NOTE: I DID NOT INCLUDE NON-LINEAR DATA STRUCTURES IN THIS. BE SURE TO COVER THE IMPORTANT CONCEPTS OF TREES.