

Complete Interview Preparation



Detailed
Course Syllabus

1) Resume Building

Resume Building

2) Programming Languages

Programming Languages:

- **C++ :** Introduction and Basic I/O, Variables, Different Errors, Operators, Loops, Arrays, String, Functions, Pointers, Dynamic Memory Allocation, Exception Handling and Smart Pointers
- **Java :** Introduction and Basic I/O, Variables , Operators, Loops, Exception Handling, Arrays, String , Immutable Strings, ArrayList , BigInteger

3) Object Oriented Programming

Object Oriented Programming:

- Classes and Objects
- Inheritance and Polymorphism : Overloading and Overriding
- Abstraction and Encapsulation
- Access Modifiers
- Friend and Virtual functions in C++
- static, final, this and super keywords and Interfaces in Java

4) Data Structures (Basics)

- **Analysis of Algorithms:**
 - Growth of functions
 - Asymptotic Notations Omega, Theta,
 - Recursion Tree Method
 - Space Complexity
- **Arrays:**
 - Insertion, Deletion, Updation, Shifting
 - Reversal, Sort Check, Maximum, Minimum
- **Recursion**
 - Introduction to Recursion
 - Tail Recursion
 - Natural Number Check Using Recursion

- Palindrome Check Using Recursion
 - Sum of Digits, Rod Cutting and Subsets
 - Tower of Hanoi
- **Hashing:**
 - Introduction to Hashing
 - Direct Address Table
 - Collision Handling
 - Chaining
 - Open Addressing
 - Double Hashing
 - Chaining Vs Open Addressing
- **String:**
 - Introduction to Strings
- **Searching:**
 - Linear Search
 - Binary Search (Iterative and Recursive)
- **Sorting:**
 - Stability in Sorting Algorithm
 - Bubble Sort
 - Selection Sort
 - Insertion Sort
 - Quick Sort
 - Different Partition Schemes in QuickSort
 - Merge Sort
 - Lomuto Partition
 - Hoare Partition
 - Heap Sort
 - Counting Sort
 - Radix Sort
 - Bucket Sort
- **Linked List:**
 - Drawback of Arrays
 - Introduction to Linked List and Implementation
 - Traversal, Insertion and Deletion
 - Sorted Insertion in Linked List
 - Reversal of Linked List (Iterative and Recursive)
 - Finding Middle
 - Remove Duplicate from Sorted Linked List
- **Circular Linked List:**
 - Traversal
 - Insertion (Head, End)
 - Deletion (Head, Kth Node)

- **Doubly Linked List:**
 - Traversal
 - Insertion (Head, End)
 - Deletion (Head, End)
 - Reversal
 - Circular Doubly Linked List
- **Stack:**
 - Introduction to Stack Data Structure
 - Implement using array
 - Implementation using Linked List
 - Stack Applications
- **Queue:**
 - Introduction to Queue Data Structure
 - Implementation using array
 - Implementation using Linked List.
- **Dequeue:**
 - Introduction to Deque Data Structure.
 - Implementations using Array
 - Implementation using Linked List
- **Tree:**
 - Implementation
 - Traversals: preorder, postorder, inorder, level order(Iterative & Recursive)
 - Binary Tree: Height, Size, Maximum
 - Print Nodes at K Distance
- **BST:**
 - Implementation
 - Search
 - Insertion
 - Deletion
 - Floor and Ceil in BST in CPP and Java
 - Self Balancing BST
 - AVL Tree (Introduction and applications)
 - Red-Black Tree (Introduction and applications)
 - Applications of BST
- **Heap:**
 - Implementation
 - Insert
 - Heapify and Extract in Heap
 - Decrease Key, Delete and Build Heap

5) Libraries

C++ STL

- **Introduction to STL**
 - i) Introduction and Application
 - ii) Iterators
 - iii) Templates
 - iv) Function Templates
 - v) Class Templates
- **Pairs in CPP STL**
 - i) Introduction
 - ii) Problem(With Video Solutions): Sorting an array according to another array
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Vectors in CPP STL**
 - i) Introduction
 - ii) Vector Declaration
 - iii) More functions of Vectors
 - iv) Time Complexities of different operations and passing Vectors to function
 - v) Internal Working of Vectors
 - vi) Problems(With Video Solutions):
 - (1) Vector and Vector of Pairs
 - (2) Keeping track of previous indexes after sorting a Vector
- **Forward_list and list**
 - i) Forward List in C++ STL
 - ii) List in C++ STL
 - iii) Problems(With Video Solutions):
 - (1) Josephus Problem using List in STL
 - (2) Design a Data Structure with Insert/Replace/Print operations
 - iv) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Deque**
 - i) Introduction

- ii) Problems(With Video Solutions):
 - (1) Sliding Window Maximum
 - (2) Design a Data Structure with Min/Max operations in $O(1)$ time
- iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Stack**
 - i) Introduction and Various Operations
 - (1) push()
 - (2) pop()
 - (3) top()
 - (4) size()
 - (5) empty()
 - ii) Problems(With Video Solutions):
 - (1) Reverse items using Stack
 - (2) Balanced Parenthesis
 - (3) Stock Span Problem
 - (4) Previous Greater Elements
 - (5) Next Greater Elements
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Queue**
 - i) Introduction and Various Operations
 - (1) push()
 - (2) pop()
 - (3) front()
 - (4) back()
 - (5) empty()
 - (6) size()
 - ii) Problems(With Video Solutions):
 - (1) Reverse first K items in a Queue
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Priority Queue**
 - i) Introduction and Various Operations
 - (1) push()

- (2) `pop()`
 - (3) `top()`
 - (4) `empty()`
 - (5) `size()`
 - (6) Creating Min Heap based Priority Queue
 - ii) Problems(With Video Solutions):
 - (1) Sort an array using Priority Queue
 - (2) K Largest Elements in an array
 - (3) Buy maximum items with given money
 - (4) Find K most frequent elements
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Set & MultiSet**
 - i) Set in C++ STL
 - (1) Introduction and Implementation
 - (2) `insert()`
 - (3) `begin()`
 - (4) `end()`
 - (5) `rbegin()`
 - (6) `rend()`
 - (7) `erase()`
 - (8) `clear()`
 - (9) `find()`
 - (10) Internal Working
 - (11) Time Complexities
 - ii) Problems on Set(With Video Solutions):
 - (1) Design a Data Structure that supports the below operations:
 - (2) `insert()`
 - (3) `delete()`
 - (4) `search()`
 - (5) `getFloor()`
 - (6) `getCeiling()`
 - iii) Multiset in C++ STL with few operations
 - iv) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Map and MultiMap**
 - i) Introduction to Map

- (1) insert()
 - (2) operator()
 - (3) size()
 - (4) empty()
 - (5) clear()
 - (6) begin()
 - (7) end()
 - (8) Internal Working
 - (9) Time Complexities
- ii) Problem:
 - (1) Design a data structure for item prices. The operations are add(), find(), findGreater(), findSmaller() and printSorted()
 - (2) Count greater elements for every array element.
- iii) Multimap in C++ STL with few functional operations
- iv) Problem(With Video Solutions):
 - (1) Design a data structure for prices with duplicates allowed. The operations are add(), find(), findGreater(), findSmaller() and printSorted
- v) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Unordered_set**
 - i) Introduction to Set
 - (1) insert()
 - (2) begin()
 - (3) size()
 - (4) end()
 - (5) clear()
 - (6) find()
 - (7) Internal Working
 - (8) Time Complexities
 - ii) Problems(With Video Solutions):
 - (1) Print Unique Elements of Array
 - (2) Print duplicate elements of the array
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- **Unordered_Map**
 - i) Introduction
 - ii) Problems(With Video Solutions):

- (1) Design a DS for storing user balance
 - (2) Find Winner of Election
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- o **Non Mutating STL Algorithms**
 - i) Explanation along with Time Complexities of
 - (1) max_element()
 - (2) min_element()
 - (3) accumulate()
 - (4) count()
 - (5) find()
 - (6) binary_search()
 - (7) lower_bound()
 - (8) upper_bound()
 - (9) rotate()
 - (10) fill()
 - (11) is_permutation()
 - (12) rand()
 - ii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.
- o **Mutating STL Algorithm**
 - i) Explanation along with Time Complexities of
 - (1) sort()
 - (2) reverse()
 - (3) next_permutation()
 - (4) prev_permutation()
 - (5) make_heap()
 - (6) merge()
 - ii) Problems(With Video Solutions):
 - (1) The Thief problem
 - (2) Fractional knapsack problem
 - (3) Chocolate Distribution problem
 - (4) Sort array elements by frequency
 - iii) Practice Problems
 - (1) This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

Java Collections

- **Collection Overview**
 - i) Introduction to Java Collections Framework
 - ii) Collections hierarchy
 - iii) Generics
 - iv) Wildcards
 - v) toArray() Methods
 - vi) Collections Interface
 - vii) Iterators
 - viii) Collections Bulk operations
 - ix) Iterating through Collections
- **Java Lambda Expressions**
 - i) Introduction to Lambda Expressions and ways to use them
 - ii) Introduction to Method References and examples
 - iii) Syntax of Lambda Expressions
 - iv) Practice Problems
 - (1) Practice problems on Lambda Expressions
- **Java Streams**
 - i) Introduction to Streams in Java
 - ii) Various Applications of Streams
 - iii) The Stream hierarchy and methods
 - iv) Examples on Streams
 - v) Practice Problems
 - (1) Practice problems on Streams
- **ArrayList**
 - i) Introduction to List Interface
 - ii) Using List Iterator
 - iii) Introduction to ArrayLists
 - iv) Implementation
 - v) ArrayList Methods
 - vi) Traversal
 - vii) Problems with video explanation
 - (1) List of smaller elements
 - viii) Practice Problems
 - (1) Practice problems on implementation, iterator, methods, and using ArrayList to solve dsa problems
- **Linked List**
 - i) Introduction and implementation of LinkedList in Java
 - ii) Problems with video explanation

- (1) Josephus Problem using LinkedList
 - (2) Design a DS for remove and print
 - iii) Practice Problems
 - (1) Practice problems on implementation, traversal, and use of LinkedList,
- **Stack**
 - i) Introduction to Stack
 - ii) Implementation
 - iii) Methods
 - iv) Traversal
 - v) Problems with video explanation
 - (1) Reverse order of items
 - (2) Check for balanced parentheses
 - (3) Stock span
 - (4) Previous greater element
 - (5) Next greater element
 - vi) Practice Problems
 - (1) Practice problems on implementation, methods, and using Stacks to solve dsa problems
- **Queue**
 - i) Introduction to Queue Interface
 - ii) Implementation and usage
 - iii) Methods
 - iv) Traversal
 - v) Problems with video explanation
 - (1) Reverse first k items
 - vi) Practice Problems
 - (1) Practice problems on implementation, methods, and using Queue to solve dsa problems
- **Deque**
 - i) Introduction to Deque
 - ii) Implementation and usage
 - iii) ArrayDeque
 - iv) Methods
 - v) Traversal
 - vi) Practice Problems
 - (1) Practice problems on implementation, methods, and using ArrayDeque to solve dsa problems
- **PriorityQueue**
 - i) Introduction to PriorityQueue
 - ii) Implementation and usage
 - iii) Methods

- iv) Traversal
- v) Problems with video explanation
 - (1) Purchasing maximum items
 - (2) K largest elements
 - (3) Find k most frequent
 - (4) Find k most frequent in Linear time
- vi) Practice Problems
 - (1) Practice problems on implementation, methods, and using PriorityQueue to solve dsa problems
- **HashSet and LinkedHashSet**
 - i) Introduction to HashSet
 - ii) Introduction to LinkedHashSet
 - iii) Implementation and usage
 - iv) Methods
 - v) Traversal
 - vi) Problems with video explanation
 - (1) Print distinct elements
 - (2) Print repeating elements
 - vii) Practice Problems
 - (1) Practice problems on implementation, methods, and using HashSet to solve dsa problems
- **TreeSet**
 - i) Introduction to TreeSet
 - ii) Implementation and usage
 - iii) Methods
 - iv) Traversal
 - v) Problems with video explanation
 - (1) Ceiling on right
 - (2) Count greater element
 - vi) Practice Problems
 - (1) Practice problems on implementation, methods, and using TreeSet to solve dsa problems
- **HashMap and LinkedHashMap**
 - i) Introduction to HashMap
 - ii) Introduction to LinkedHashMap
 - iii) Implementation and usage
 - iv) Methods
 - v) Traversal
 - vi) Problems with video explanation
 - (1) DS for balance
 - (2) Print frequencies in order
 - vii) Practice Problems

- (1) Practice problems on implementation, methods, and using HashMap to solve dsa problems
- **TreeMap**
 - i) Introduction to TreeMap
 - ii) Implementation and usage
 - iii) Methods
 - iv) Traversal
 - v) Problems with video explanation
 - (1) Design a data structure for item prices
 - (2) Design a data structure for item prices with duplicates allowed
 - vi) Practice Problems
 - (1) Practice problems on implementation, methods, and using TreeMap to solve dsa problems
- **String**
 - i) Introduction to Strings
 - ii) Introduction to StringBuilder and StringBuffer
 - iii) Implementation and usage
 - iv) Methods
 - v) Traversal
 - vi) Problems with video explanation
 - (1) Pangram checking
 - (2) Pattern searching
 - (3) Find one extra character
 - vii) Practice Problems
 - (1) Practice problems on implementation, methods, and using Strings to solve dsa problems
- **Comparator and Comparable**
 - i) Introduction to Comparable Interface
 - ii) Introduction to Comparator Interface
 - iii) Methods of Comparator Interface and Examples on it
 - iv) Practice Problems
 - (1) Practice problems on using Comparator to sort effectively
- **Arrays Class**
 - i) Introduction to Arrays and the Arrays Class
 - ii) Implementation and usage
 - iii) Methods like
 - (1) fill()
 - (2) BinarySearch()
 - (3) equals()
 - (4) mismatch()
 - (5) compare()

- (6) `asList()`
 - (7) `toString()`
 - iv) Traversal
 - v) Practice Problems
 - (1) Practice problems on implementation and methods
- **Collections Class**
 - i) Introduction to Collections Class
 - ii) Methods like `fill()`, `reverse()`, `binarySearch()`, `max()`, `min()`, `frequency()`
 - iii) Practice Problems
 - (1) Practice problems on methods
- **Sorting**
 - i) Introduction to sorting in Java
 - ii) `Arrays.sort()`
 - iii) `Collections.sort()`
 - iv) Comparable Interface
 - v) Problems with video explanation
 - (1) The thief problem
 - (2) Chocolate distribution problem
 - (3) Keep indices after sorting
 - (4) Sort an array according to other
 - (5) Sort students by marks
 - (6) Sort elements by frequency
 - (7) Sort elements by frequency in Linear Time
 - vi) Practice Problems
 - (1) Practice problems on various sorting algorithms, and comparator sort

6) Data Structures (Advanced)

- **Mathematics**
 - Count Digits
 - Palindrome Numbers
 - Factorial of Numbers
 - GCD of Two Numbers
 - LCM of Two Numbers
 - Check for Prime
 - Prime Factors
 - Sieve of Eratosthenes
 - Computing Power

- **Bit Magic**
 - Bitwise Operators in CPP(Part 1)
 - Bitwise Operators in CPP(Part 2)
 - Bitwise Operators in JAVA(Part 1)
 - Bitwise Operators in JAVA(Part 2)
 - Bitwise Operators in JAVA(Part 3)
 - Check Kth bit is set or not
 - Count set bits
 - Power of Two
 - One Odd Occurring
 - Two Odd Occurring
 - Power Set using Bitwise
- **Recursion**
 - Josephus Problem
 - Subset Sum Problem
- **Arrays:**
 - Kadane's Algorithm
 - Shuffling Algorithms
 - Sliding Window
 - Prefix Sum Technique
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Matrix:**
 - Multidimensional Array in CPP and Java
 - Search, Transpose and Rotate
 - Pattern Traversal: Snake, Spiral, Boundary
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Searching:**
 - Two Pointer Approach
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Sorting:**
 - Union And Intersection of Sorted Arrays
 - Inversions Count
 - Tail Call elimination Quick Sort
 - Cycle Sort
 - Merge of Overlapping Intervals
 - Overview of Sorting Algorithms
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.

- **Hashing:**
 - Double Hashing
 - Find frequencies of array
 - Count Distinct element in Every Window
 - Intersection and Union via Hashing
 - Frequencies of Array Elements
 - Distinct Elements in Window
 - Counting Occurrences
 - Check for a Pair with given Sum
 - Longest Consecutive Subsequence
 - Subsequence Problems
 - Subarray Problems
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Strings:**
 - Creation, Updation
 - Reverse, Pangram, Case conversion
 - Validation, Length
 - Palindrome Check
 - Overview of Pattern Searching
 - Pattern Matching Algorithms:
 - Rabin Karp Algorithm
 - KMP Algorithm
 - Rotations Check of two Strings
 - Anagram
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Linked List:**
 - Doubly Linked List
 - Circular Linked List
 - Loop in Linked List (Detection and Removal)
 - Loop Detection Algorithms
 - Union and Intersection of LinkedLists
 - Reverse in Groups
 - LRU Cache Design
 - Palindrome LinkedList
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Stack:**
 - Infix, Postfix, Prefix (Introduction)
 - Infix to PostFix (Simple Solution)
 - Infix to PostFix (Efficient Solution)
 - Evaluation of Postfix
 - Infix to Prefix (Simple Solution)

- Infix to Postfix (Efficient Solution)
 - Evaluation of Prefix
 - Implementing Two Stacks in Single Array
 - Implementing K stacks in Single Array
 - Largest Rectangular Area in Histogram
 - Design a Stack that supports getMin() operation
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Queue and Deque:**
 - Stack using Queue
 - Reversal
 - Maximum of all Subarrays of Size K
 - Generate numbers using given digits
 - Design a data structure with min/max operations
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Tree:**
 - Line By Line Level Order Traversal
 - Printing Left, Right, Top and Bottom Views
 - Binary Tree to Doubly Linked List
 - Binary Tree from Inorder and Postorder Traversal
 - Maximum Width
 - Child Sum Property
 - Convert Binary Tree to Doubly LinkedList
 - Burning a Tree from Leaf
 - Diameter
 - LCA
 - Serialize and Deserialize
 - Count Nodes in Complete Binary Tree
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Binary Search Tree:**
 - Top View
 - Bottom View
 - Vertical Sum
 - Vertical Traversal
 - Fix BST With Two Nodes Swapped
 - Check For BST
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Heap:**
 - Heap Sort
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.

- **Graph:**
 - Graph Representation: Adjacency List
 - Adjacency List Implementation in CPP
 - Adjacency List Implementation in Java
 - Adjacency List and Matrix Comparison
 - Breadth First Search and application
 - Depth First Search and application
 - Detect Cycle in Undirected Graph
 - Detect Cycle in Directed Graph
 - Topological Sorting
 - Shortest Path Problems
 - Prim's Algorithm Introduction and Implementation in CPP and Java
 - Dijkstra's Algorithm Introduction and Implementation in CPP and Java
 - Bellman Ford Algorithm
 - Kosaraju's Algorithm
 - Articulation Point
 - Bridges in Graph
 - Tarjan's Algorithm
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Greedy Algorithm:**
 - Introduction
 - Activity Selection Problem in CPP and Java
 - Fractional Knapsack in CPP and Java
 - Job Sequencing Problem
 - Huffman Coding
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **BackTracking:**
 - Concept of Backtracking
 - Problems: Rat In Maze, N Queen, Sudoku
 - More Problems for Practice.
- **Dynamic Programming:**
 - Introduction
 - Memoization
 - Tabulation
 - LCS and its variations
 - Coin Change
 - KnapSack
 - LIS and its variations
 - Egg Drop Puzzle
 - Subset Sum
 - Matrix Chain Multiplication
 - Palindrome Partitioning

- Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Trie:**
 - Introduction
 - Insert, Search, Delete
 - Video Solutions for some standard and complex problems
 - More Problems for Practice.
- **Segment Tree:**
 - Introduction
 - Construction
 - Range and Update Query
 - More Problems for Practice.
- **Disjoint-Set**
 - Introduction
 - Union-Find
 - Union By Rank
 - Path Compression
 - Kruskal's Algorithm
 - More Problems for Practice

7) Object Oriented Analysis and Design

Object oriented Analysis and Design Concepts

- Introduction to Object and Classes.
- Software Development Process.
- UML and its importance.
- Class Diagrams & Object Diagrams
- Use-case Diagrams

Object-oriented Analysis and Design Case Studies: Complete step by step design and analysis of below case studies.

- BookMyShow : Movie ticket booking application.
- MyFlipCart : Complete e-commerce application.
- ParkingLot : Automated Solution for Parking-Lots.
- BlackJack : Most popular card game in casinos.

8) CS Subjects

- **Operating System**
- Operating System and its Types
- Multiprogramming, Multiprocessing, Multithreading
- Process Management and Scheduling
- Process Synchronization.
- Deadlock
- Memory Management
- **Database Management System**
- Introduction to DBMS
- Architectures
- ER Model
- Relational Model
- Keys in Relational Model
- Database Normalization
- Normal Forms
- Concurrency Control
- Indexing in Database
- B+ Tree Introduction
- SQL
- **Computer Networks:**
- Introduction to Computer Networks
- TCP/IP vs OSI Model
- Circuit Switching vs Packet Switching
- Flow Control Protocols
- IP and Classful Addressing
- Classless Addressing
- Routing Protocols
- ARP & DHCP
- Transport Layer
- TCP & UDP
- Application Layer
- **Subject Wise Most Asked Interview Questions**
- Virtual Memory

9) Aptitude and Reasoning

QUANTITATIVE ANALYSIS

- **Number System** - Numbers, Prime & Composite Numbers, Co-Prime numbers,
- **Divisibility Test** - Divisibility, Factor, Prime Factor, Divisibility Rules(2, 3, 4,, 10, 11)

- **HCF and LCM** - Listing Multiples, Prime Factorization, Division method, etc.
- **Decimals Fractions** - Fractions, Decimals, Decimal Fractions, Recurring Decimals.
- **Squares & Cubes** - Square, Cube, Square Root, Cube Root
- **Average** - Mean, Median and Mode
- **Age** - Various techniques to solve age problems.
- **Log** - Log Function, Common Log, Natural Log, Binary Log, Laws of Logarithms.
- **Percentage** - Percentage, Fractions of Percentages, Expenditure, Price, Consumption, Population, Depreciation, ...
- **Profit, Loss & Discount** - CP, SP, MP, Profit, Loss, Discount, ...
- **Ratio And Proportion** - Ratio, Proportion, Compounded Ratio, Mean Proportional, Componendo, Dividendo, Directly Proportional, Inversely Proportional.
- **Partnership** - Various types of Partnership and Partners.
- **Pipe and Cistern | Part 1**
- **Pipe and Cistern | Part 2**
- **Time And Work** - Problems on Time, Work and Efficiency.
- **Work and Wages | Part 1**
- **Work and Wages | Part 2**
- **Speed, Distance and Time | Part 1**
- **Speed, Distance and Time | Part 2**
- **Boats and Streams** - Downstream, Upstream, Average Speed, ...
- **Trains** - Problems in same and opposite Direction.
- **Simple Interest**
- **Compound Interest | Part 1**
- **Compound Interest | Part 2**
- **Area** - Rectangle, Square, Triangle etc.
- **Volume** - Cube, Cuboid, Cylinder, Cone, Sphere, Hemisphere,
- **Race** - Race, Winner, Dead Heat Race, etc.
- **Clocks** - Problems related to angle between hands.
- **Calendars** - Day, Week, Month, Year, Leap Year, Non-Leap year, Odd days etc.
- **Height Distance** - Heights, Height, Distance, Angle of Elevation, Depression, Trigonometry Ratio, Conversion
- **Series And Sequence** - AP, GP, HP.

LOGICAL and VERBAL REASONING

- **LOGICAL REASONING** -
 - Introduction,
 - Data Sufficiency,
 - Data Interpretation,
 - Blood Relations,
 - Sequence and Series,

- Direction Test,
 - Mathematical Operations,
 - Syllogism.
- **VERBAL APTITUDE -**
 - Verbal Ability
 - Reading Comprehension
 - Vocabulary Section
 - Vocabulary Reasoning Section
- **BASICS OF GRAMMAR 1 -**
 - Section
 - Clause
 - Phrase
 - Parts of Speech
 - Nouns
 - Gerunds
 - Pronouns
 - Verbs
 - Adjectives.
 - Adverbs
- **BASICS OF GRAMMAR 2 -**
 - Prepositions
 - Conjunctions
 - Subordinating Conjunctions
 - Tenses
 - Interjections
- **ARTICLE -**
- **ACTIVE VOICE & PASSIVE VOICE -**
 - Use with Tenses
- **CLOSET TESTS -**
 - Problem Solving
- **PASSAGE FORMATION -**
 - Problem Solving
- **SENTENCE FORMATION -**
 - Problem Solving
- **SENTENCE COMPLETION-**
 - Problem Solving
- **SUBJECT VERB and AGREEMENT -**
 - General Rule
 - Multiple Subjects
 - Inverted Sentences
 - There is & There are
 - Collective Nouns
 - Or and Nor
 - Indefinite Pronouns

- Number Of and Percentage Of
- **DETERMINERS**
 - Demonstration and Use
- **MODIFIERS**
 - General Rules
 - Problem Solving
 - Dangling Modifiers
- **PARALLEL STRUCTURE**
 - General Rules
 - Problems with Articles, Infinitives, and Prepositions.
 - Conjunction Pairs
 - Comparing and Contrasting
- **GRAMMAR EXERCISE -**
 - Subject-Verb Agreement
 - Pronouns
 - Verbs
 - Parallel Structure
 - Modifiers
 - Adjectives and Adverbs
- **ERROR SPOTTING**
 - Types of Errors
 - Problem Solving
- **PARAJUMBLES**
 - Problem Solving
- **VERBAL ANALOGIES**
 - Problem Solving