Comprehensive Programming & App Development Interview Question Bank

Total Questions: 200

Part 1: Computational Aptitude & Logical Reasoning (Questions 1-40)

1. A train 240m long passes a pole in 24 seconds. How long will it take to pass a platform 650m long?

Answer: Speed = Distance/Time = 240/24 = 10 m/s. Total distance for platform = Train + Platform = 240 + 650 = 890m. Time = 890/10 = 89 seconds.

2. If A can do a work in 10 days and B in 15 days, how long will they take together?

Answer: 1/10 + 1/15 = (3+2)/30 = 5/30 = 1/6. It takes 6 days.

3. What is the angle between hands of a clock at 3:40?

Answer: Hour hand moves 0.5°/min. At 3:40, Hour hand is at 90° + (400.5) = 110°. Minute hand is at 406 = 240°. Diff = 240 - 110 = 130°.

4. Find the missing number: 2, 6, 12, 20, 30, ?

Answer: 42. The difference increases by 2 each time (+4, +6, +8, +10, +12).

5. In a race of 1km, A beats B by 100m or 10s. What is A's time?

Answer: B runs 100m in 10s -> Speed of B = 10m/s. B runs 1000m in 100s. A beats B by 10s, so A takes 100 - 10 = 90s.

6. Probability of getting a sum of 9 with two dice?

Answer: Pairs: (3,6), (4,5), (5,4), (6,3). Total outcomes = 36. Prob = 4/36 = 1/9.

7. A father is 3 times as old as his son. After 10 years, he will be twice as old. Find present ages.

Answer: $F = 3S$. $(F+10) = 2(S+10) \Rightarrow 3S+10 = 2S+20 \Rightarrow S=10$. Father is 30.

8. If "CAT" is coded as 3120, how is "DOG" coded?

Answer: C=3, A=1, T=20. D=4, O=15, G=7. Code: 4157.

9. Pointing to a man, a woman said, "His mother is the only daughter of my mother." Who is the man to the woman?

Answer: The "only daughter of my mother" is the woman herself. So, "His mother is [Me]". The man is her Son.

10. What is 20% of 30% of 500?

Answer: $0.2 \times 0.3 \times 500 = 0.06 \times 500 = 30$.

11-15. Logical Syllogisms:

11. All Cats are Dogs. Some Dogs are Birds. Answer: No definite conclusion that Cats are Birds.

12. No A is B. All B are C. Answer: Some C are not A.

13. Some papers are pens. All pens are scales. Answer: Some scales are papers.

14. All cars are trucks. All trucks are cycles. Answer: All cars are cycles.

15. Some trees are flowers. Some flowers are fruits. Answer: No relation between Trees and Fruits.

16. Permutation: In how many ways can letters of "APPLE" be arranged?

Answer: 5 letters, 2 Ps. $5! / 2! = 120 / 2 = 60$.

17. Combination: How many ways to select 2 students from 5?

Answer: $5C2 = (5 \times 4) / 2 = 10$.

18. Simple Interest: Principal=1000, Rate=10%, Time=2 years.

Answer: $SI = (P \times R \times T)/100 = 200$.

19. Compound Interest: Principal=1000, Rate=10%, Time=2 years.

Answer: $Amount = 1000(1.1)^2 = 1210$. $CI = 210$.

20. Profit & Loss: Cost Price = 100, Selling Price = 120. Profit %?

Answer: $(20/100) \times 100 = 20\%$.

21-30. Quick Math & Logic:

21. What is the next prime after 17? 19

22. $LCM(4, 6)$? 12

23. $HCF(12, 18)$? 6

24. If $x + 1/x = 2$, find x. 1

25. Sum of first 10 natural numbers? $n(n+1)/2 = 55$

26. Find the odd one out: 27, 64, 100, 125. 100 (Others are perfect cubes).

27. 3 cats kill 3 rats in 3 minutes. Time for 100 cats to kill 100 rats? 3 minutes.

28. Binary of 10? 1010

29. Average of 10, 20, 30? 20

30. If today is Monday, what day is it after 64 days? Tuesday (64%7 = 1 remainder).

31-40. Data Interpretation (General):

31. How to calculate growth percentage? ((New-Old)/Old) * 100.

32. What is a Pie Chart used for? Representing parts of a whole.

33. How to convert km/hr to m/s? Multiply by 5/18.

34. Volume of a Cube? Side³

35. Area of a Circle? πr^2

36. Speed of stream question formula? Downstream = U+V, Upstream = U-V.

37. Probability of Heads in a coin toss? 0.5

38. Value of $5!$? 120

39. Square root of 225? 15

40. Cube root of 27? 3

Part 2: PDSA - Data Structures & Algorithms (Questions 41-90)

41. What is a Data Structure?

Answer: A way of organizing and storing data efficiently so it can be accessed and modified.

42. Array vs. Linked List?

Answer: Array: Fixed size, fast access ($O(1)$), slow insertion ($O(n)$). Linked List: Dynamic size, slow access ($O(n)$), fast insertion ($O(1)$).

43. What is a Stack?

Answer: LIFO (Last In First Out) structure. Operations: Push, Pop, Peek.

44. What is a Queue?

Answer: FIFO (First In First Out) structure. Operations: Enqueue, Dequeue.

45. Explain a Hash Table.

Answer: Stores key-value pairs. Uses a hash function to compute an index. Fast lookup ($O(1)$).

46. What is a Collision in Hashing?

Answer: When two keys hash to the same index. Handled via Chaining (Linked Lists) or Open Addressing.

47. What is a Binary Search Tree (BST)?

Answer: A tree where Left Child < Parent < Right Child. Allows binary search.Shutterstock

48. Time Complexity of Binary Search?

Answer: $O(\log n)$ (Best/Average).

49. Time Complexity of Linear Search?

Answer: $O(n)$.

50. What is Recursion?

Answer: A function calling itself with a base case to terminate.

51-60. Sorting & Searching:

51. Bubble Sort Complexity? $O(n^2)$.

52. Merge Sort Complexity? $O(n \log n)$.

53. Quick Sort Pivot? Element used to partition the array.

54. Stable Sort? Preserves relative order of equal elements.

55. Best Sorting Algo for large data? Merge Sort or Quick Sort.

56. Insertion Sort? Good for small or nearly sorted arrays.

57. Selection Sort? Selects min element and swaps to front.

58. Why is QuickSort preferred over MergeSort for arrays? Uses less memory (in-place).

59. Binary Search precondition? Array must be sorted.

60. What is an In-place algorithm? Uses constant extra space $O(1)$.

61-70. Trees & Graphs:

61. In-order Traversal? Left -> Root -> Right.

62. Pre-order Traversal? Root -> Left -> Right.

63. Post-order Traversal? Left -> Right -> Root.

64. Height of a Tree? Longest path from root to leaf.

65. BFS (Breadth First Search)? Uses a Queue. Level by level.

66. DFS (Depth First Search)? Uses a Stack (Recursion). Deepest path first.

67. Graph representation? Adjacency Matrix or Adjacency List.

68. Leaf Node? A node with no children.

69. Root Node? The top node of a tree.

70. Balanced Tree? Height difference between subtrees is minimal (e.g., AVL Tree).

71-80. Advanced DS:

71. Priority Queue? Elements processed based on priority, not arrival.

72. Heap (Min/Max)? A tree where parent is always smaller/larger than children.

73. Graph Cycle Detection? Use DFS and check for visited nodes in current path.

74. Dijkstra’s Algorithm? Finds shortest path in weighted graphs.

75. Stack Overflow? Occurs when recursion is too deep (memory full).

76. Circular Queue? Last position connects back to the first.

77. Doubly Linked List? Nodes have pointers to both Next and Previous.

78. Dynamic Array (ArrayList)? Resizes itself when full (usually doubles capacity).

79. Sparse Matrix? A matrix with mostly zero values.

80. Greedy Algorithm? Makes the locally optimal choice at each step.

81-90. Big O Notation (Complexity):

81. $O(1)$? Constant time (Accessing array index).

82. $O(n)$? Linear time (Looping once).

83. $O(n^2)$? Quadratic time (Nested loops).

84. $O(\log n)$? Logarithmic (Halving input, Binary Search).

85. Space Complexity? Memory required by algorithm.

86. Worst Case? Max time an algorithm takes.

87. Best Case? Min time an algorithm takes.

88. Average Case? Expected time on random input.

89. Complexity of accessing Hash Map? $O(1)$ average.

90. Complexity of Fibonacci (Recursive)? $O(2^n)$ (Exponential - very bad).

Part 3: Java Basics & OOP (Questions 91-140)

91. What is JVM?

Answer: Java Virtual Machine. Executes bytecode. Makes Java platform-independent.

92. JDK vs JRE?

Answer: JDK = JRE + Dev Tools (Compiler). JRE = JVM + Libraries (Runner).

93. What is Bytecode?

Answer: Compiled Java code (.class file) that runs on JVM.

94. Four Pillars of OOP?

Answer: Encapsulation, Abstraction, Inheritance, Polymorphism.

95. Class vs Object?

Answer: Class is a blueprint; Object is an instance of the class.

96. What is a Constructor?

Answer: Special method to initialize objects. Same name as class, no return type.

97. static keyword?

Answer: Belongs to the class, not instance. Shared memory.

98. final keyword?

Answer: Variable = Constant. Method = Cannot override. Class = Cannot inherit.

99. this keyword?

Answer: Refers to the current object instance.

100. super keyword?

Answer: Refers to the parent class object.

101-110. OOP Concepts:

101. Method Overloading? Same name, different parameters (Compile-time polymorphism).

102. Method Overriding? Same name/params in Child class (Runtime polymorphism).

103. Inheritance? Acquiring properties of parent class (extends).

104. Encapsulation? Hiding data using private and exposing via getters/setters.

105. Abstraction? Hiding implementation details (Abstract classes/Interfaces).

106. Interface? Pure abstraction. All methods public abstract (pre-Java 8).

107. Abstract Class? Can have both abstract and concrete methods.

108. Can Java support Multiple Inheritance? No, but possible via Interfaces.

109. Polymorphism? "Many forms". Treating child objects as parent type.

110. Dynamic Binding? Method call resolved at runtime (Overriding).

111-120. Core Java:

111. String Immutability? Once created, cannot change. Secure and thread-safe.

112. StringBuffer vs StringBuilder? Buffer is synchronized (Slow/Safe). Builder is not (Fast/Unsafe).

113. equals() vs ==? equals compares content; == compares reference address.

114. Wrapper Classes? Object representation of primitives (int -> Integer).

115. Autoboxing? Automatic conversion of primitive to Wrapper.

116. public static void main(String args[])? Entry point of program.

117. Package? Group of related classes.

118. Access Modifiers? Private, Default, Protected, Public.

119. Garbage Collection? Automatic memory management (removes unused objects).

120. finalize()? Method called before GC destroys object.

121-130. Exception Handling:

121. Exception? Runtime error disrupting flow.

122. Try-Catch? Try block contains code; Catch handles exception.

123. Finally? Executes always (cleanup).

124. Throw vs Throws? Throw = explicitly throw exception. Throws = declare method might throw.

125. Checked Exception? Compile-time (IOException).

126. Unchecked Exception? Runtime (NullPointerException).

127. Custom Exception? Extending Exception class.

128. Error vs Exception? Error (StackOverflow) is unrecoverable. Exception is recoverable.

129. Hierarchy? Throwable -> Exception/Error.

130. Try with Resources? Auto-closes resources (Java 7+).

131-140. Collections & Threads:

131. ArrayList? Dynamic array. Fast access.

132. LinkedList? Doubly linked list. Fast insert/delete.

133. HashMap? Key-Value pair storage.

134. HashSet? Unique elements only.

135. HashTable vs HashMap? Table is synchronized (old); Map is not.

136. Thread? Lightweight sub-process.

137. How to create Thread? Extend Thread or implement Runnable.

138. Life cycle of Thread? New -> Runnable -> Running -> Blocked -> Dead.

139. Synchronization? Preventing thread interference.

140. Iterator? Interface to traverse collections.

Part 4: Modern App Development (Questions 141-200)

Section A: HTML & CSS (141-160)

141. What is HTML?

Answer: HyperText Markup Language. Structure of web.

142. Semantic Tags?

Answer: Tags with meaning (<header>, <article>, <footer> vs).

143. Block vs Inline elements?

Answer: Block starts new line (div, p). Inline takes necessary width (span, a).

144. What is the DOM?

Answer: Document Object Model. Tree structure of HTML elements.

145. CSS Box Model?

Answer: Content + Padding + Border + Margin.

146. id vs class?

Answer: ID is unique (#). Class is reusable (.).

147. Flexbox?

Answer: 1D layout model (Row/Column). display: flex.

148. CSS Grid?

Answer: 2D layout model (Rows and Columns).

149. position: absolute vs relative?

Answer: Relative is relative to itself. Absolute is relative to nearest positioned ancestor.

150. Responsive Design?

Answer: Web pages adapt to screen size (using Media Queries).

151. z-index?

Answer: Controls stack order of elements (which one is on top).

152. Pseudo-class?

Answer: State of element (e.g., :hover).

153. Specificity?

Answer: Hierarchy of styles (ID > Class > Tag).

154. Viewport meta tag?

Answer: Controls layout on mobile devices.

155. Bootstrap?

Answer: CSS framework for responsive design.

156. display: none vs visibility: hidden?

Answer: None removes from layout. Hidden keeps space but invisible.

157. Box-sizing: border-box?

Answer: Includes padding/border in width calculation.

158. CSS Variables?

Answer: --main-color: blue; accessed via var().

159. <link> vs <script>?

Answer: Link for CSS (head). Script for JS (end of body).

160. Alt attribute?

Answer: Alternative text for images (Accessibility/SEO).

Section B: Vue.js (161-180)

161. What is Vue.js?

Answer: Progressive JavaScript framework for building UIs.

162. Virtual DOM?

Answer: In-memory copy of DOM. Vue updates this first for performance.

163. Two-way binding?

Answer: Data syncs between Model (JS) and View (HTML) using v-model.

164. v-if vs v-show?

Answer: v-if adds/removes element. v-show toggles CSS display.

165. v-for?

Answer: Directive to loop through lists.

166. v-bind (:)?

Answer: Binds attribute to data (e.g., :src="imageURL").

167. v-on (@)?

Answer: Listens to events (e.g., @click="doSomething").

168. Lifecycle Hooks?

Answer: created, mounted, updated, destroyed.

169. Computed Properties?

Answer: Cached values derived from data. Recalculates only when dependency changes.

170. Watchers?

Answer: Functions that run when specific data changes (side effects).

171. Components?

Answer: Reusable Vue instances with custom HTML.

172. Props?

Answer: Data passed from Parent to Child.

173. $emit?

Answer: Events sent from Child to Parent.

174. Vue CLI?

Answer: Command Line Interface for scaffolding Vue projects.

175. Single File Component (.vue)?

Answer: Contains Template, Script, and Style in one file.

176. Directives?

Answer: Special tokens in markup (v-if, v-for).

177. State Management?

Answer: Managing shared data (Pinia / Vuex).

178. Interpolation?

Answer: Using {{ }} to display text.

179. Key attribute in loops?

Answer: Unique ID for Vue to track nodes efficiently.

180. Methods vs Computed?

Answer: Methods run every re-render. Computed runs only on dependency change.

Section C: Flask (Backend) & General Web (181-200)

181. What is Flask?

Answer: Micro web framework for Python.

182. Flask vs Django?

Answer: Flask is minimal/flexible. Django is full-stack/batteries-included.

183. Route?

Answer: Mapping URL to a Python function.

184. Decorator in Flask?

Answer: @app.route('/') - modifies function to handle web request.

185. Jinja2?

Answer: Templating engine for Flask (Logic in HTML).

186. WSGI?

Answer: Web Server Gateway Interface. Standard for Python web apps.

187. Request Object?

Answer: Contains data sent by client (form data, args).

188. GET vs POST?

Answer: GET requests data (URL params). POST submits data (Body).

189. HTTP Status 200?

Answer: OK (Success).

190. HTTP Status 404?

Answer: Not Found.

191. HTTP Status 500?

Answer: Internal Server Error.

192. Session vs Cookie?

Answer: Session stored on server. Cookie stored on browser.

193. JSON?

Answer: JavaScript Object Notation. Format for API data.

194. REST API?

Answer: Architectural style using HTTP methods for CRUD.

195. SQLAlchemy?

Answer: ORM (Object Relational Mapper) for database access in Python.

196. Migration?

Answer: Version control for database schema changes.

197. Postman?

Answer: Tool for testing APIs.

198. Git?

Answer: Version control system for code.

199. MVC Pattern?

Answer: Model (Data), View (UI), Controller (Logic). Flask is MTV (Model-Template-View).

200. Debug Mode?

Answer: app.run(debug=True). Auto-reloads server on code change.