**Sample Questions with their answers:**

**1) What is an Abstraction?**

Abstraction means hiding the implementation detail for simplicity. It is a good programming practice to keep implementation and interface independent. Abstraction allows doing the same. We do not need to change interface, if we are going to change the implementation

**2) What is an Abstract Class?**

A class is said to be abstract when it contains at least one pure virtual function. Instantiation is not allowed for an abstract class. And the deriving class must implement or provide definition for the pure virtual functions.

Example:

```
class Base { public:    virtual void display () = 0; };
```

**3) What is algorithm?**

Algorithm is a step by step procedure, which defines a set of instructions to be executed in certain order to get the desired output.

**4) Why we need to do algorithm analysis?**

A problem can be solved in more than one ways. So, many solution algorithms can be derived for a given problem. We analyze available algorithms to find and implement the best suitable algorithm.

**5) What is asymptotic analysis of an algorithm?**

Asymptotic analysis of an algorithm refers to defining the mathematical boundation /framing of its run-time performance. Using asymptotic analysis, we can very well conclude the best case, average case and worst case scenario of an algorithm.

**6) What are asymptotic notations?**

Asymptotic analysis can provide three levels of mathematical binding of execution time of an algorithm −

- Best case is represented by Ω(n) notation.
- Worst case is represented by O(n) notation.
- Average case is represented by Θ(n) notation.

**7) What is a data structure?**

Data structure is the way data is organized (stored) and manipulated for retrieval and access. It also defines the way different sets of data relate to one another, establishing relationships and forming algorithms.

**8) What is a linear data structure? Name a few examples.**

A data structure is linear if all its elements or data items are arranged in a sequence or a linear order. The elements are stored in a non-hierarchical way so that each item has successors and predecessors except the first and last element in the list.

Examples of linear data structures are Arrays, Stack, Strings, Queue, and Linked List.

**9) What is a stack?**

A stack is an abstract data type that specifies a linear data structure, as in a real physical stack or piles where you can only take the top item off the stack in order to remove things. Thus, insertion (push) and deletion (pop) of items take place only at one end called top of the stack, with a particular order: LIFO (Last In First Out) or FILO (First In Last Out).

**10) Where are stacks used?**

- Expression, evaluation, or conversion of evaluating prefix, postfix, and infix expressions
- Syntax parsing
- String reversal
- Parenthesis checking
- Backtracking

**11) What is a queue data structure?**

A queue is an abstract data type that specifies a linear data structure or an ordered list, using the First In First Out (FIFO) operation to access elements. Insert operations can be performed only at one end called REAR and delete operations can be performed only at the other end called FRONT.

**12) List some applications of queue data structure.**

To prioritize jobs as in the following scenarios:

- As waiting lists for a single shared resource in a printer, CPU, call center systems, or image uploads; where the first one entered is the first to be processed
- In the asynchronous transfer of data; or example pipes, file IO, and sockets
- As buffers in applications like MP3 media players and CD players
- To maintain the playlist in media players (to add or remove the songs)

**13) What operations can be performed on queues?**

- enqueue() adds an element to the end of the queue
- dequeue() removes an element from the front of the queue
- init() is used for initializing the queue
- isEmpty tests for whether or not the queue is empty
- Front is used to get the value of the first data item but does not remove it
- Rear is used to get the last item from a queue

**14) What are the advantages of the heap over a stack?**

Generally, both heap and stack are part of memory and used in Java for different needs:

- Heap is more flexible than the stack because memory space can be dynamically allocated and de-allocated as needed
- Heap memory is used to store objects in Java, whereas stack memory is used to store local variables and function call
- Objects created in the heap are visible to all threads, whereas variables stored in stacks are only visible to the owner as private memory
- When using recursion, the size of heap memory is more whereas it quickly fill-ups stack memory.

**15) Define the graph data structure?**

It is a type of non-linear data structure that consists of vertices or nodes connected by edges or arcs to enable storage or retrieval of data. Edges may be directed or undirected.

**16) What are the applications of graph data structure?**

- Transport grids where stations are represented as vertices and routes as the edges of the graph
- Utility graphs of power or water, where vertices are connection points and edges the wires or pipes connecting them
- Social network graphs to determine the flow of information and hotspots (edges and

vertices)

- Neural networks where vertices represent neurons and edges the synapses between them

**17) What is an AVL tree?**

An AVL (Adelson, Velskii, and Landi) tree is a height balancing binary search tree in which the difference of heights of the left and right subtrees of any node is less than or equal to one. This controls the height of the binary search tree by not letting it get skewed. This is used when working with a large data set, with continual pruning through insertion and deletion of data.

**18) Explain the max heap data structure.**

It is a type of heap data structure where the value of the root node is greater than or equal to either of its child nodes.

**19)** How do you find the height of a node in a tree?

The height of the node equals the number of edges in the longest path to the leaf from the node, where the depth of a leaf node is 0.

**20) Explain the Types of Data Structures?**

**There are mainly two types:**

- **Linear Data Structure**: When all of its components are organized in a proper sequence, a data structure is called linear. The components are stored in linear data structures in a non-hierarchical manner where each item has the successors and predecessors except for the first and final element.
- **Non-linear data structure**: The Non-linear data structure does not form a series, i.e. each object or entity is linked to two or more than two objects in a non-linear manner. The elements of the data are not organized within the sequential model.

**21) Discuss the Different Operations that can be carried out on Data Structures?**

Following are the different operations that are generally carried out in Data Structures:

- *Insert*– Add a new data item in the already existing set of data items.
- *Delete*– Remove an existing data item from the given data item set.
- *Traverse*– Access each piece of data precisely once so that it can be processed.
- *Search*– Figure out where the data item resides in the specified data item set.
- *Sort*– Arrange the data objects in certain order i.e. in ascending or descending order fornumerical data and in dictionary order for alphanumeric data.

**22) Convert the expression ((A + B) * C - (D - E) ^ (F + G)) to equivalent Prefix and Postfix notations.**

a. Prefix Notation: - * +ABC ^ - DE + FG

b. Postfix Notation: AB + C * DE - FG + ^ -

**23) List out few of the Application of tree data-structure?**

- The manipulation of Arithmetic expression,
- Symbol Table construction,
- Syntax analysis

**24) Classify the Hashing Functions based on the various methods by which the key value is found.**

- Direct method,
- Subtraction method,
- Modulo-Division method,
- Digit-Extraction method,
- Mid-Square method,
- Folding method,
- Pseudo-random method

**25) What are the types of Collision Resolution Techniques and the methods used in each of the type?**

- Open addressing (closed hashing), The methods used include: Overflow block.
- Closed addressing (open hashing), The methods used include: Linked list, Binary tree.

**26) What is hashing?**

Hashing is the process of converting a given key into another smaller value for O(1) retrieval time.

- This is done by taking the help of some function or algorithm which is calledas hash function to map data to some encrypted or simplified representative value which is termed as “hash code” or “hash”. This hash is then used as an index to narrow down search criteria to get data quickly.

**27) Time Complexity of Kruskal’s**

In Kruskal’s algorithm, most time consuming operation is sorting because the total complexity of the Disjoint-Set operations will be O(ElogV), which is the overall Time Complexity of the algorithm.

**28) Time Complexity of Prim’s Algorithm**

The time complexity of the Prim’s Algorithm is O((V+E)logV) because each vertex is inserted in the priority queue only once and insertion in priority queue take logarithmic time.