Q) Array vs LinkedList:  
Both linear data structures, however, an array is an index-based data structure where each element is associated with an index. On the other hand, LinkedList relies on references where each node consists of the data and the references to the previous and the next node (for Double LinkedList).

Q) The keyword this vs super:

Super: super keyword is used to access methods of the parent class

This: this is used to access methods of the current class.

Q) What is software process or Software Development Life Cycle (SDLC)?

Software Development Life Cycle, or software process is the systematic development of software by following every stage in the development process.

Q) What are the various phases of Software Development?

* Requirement Gathering
* System Analysis
* Design, Coding
* Testing
* Maintenance and Documentation in that order.

Q) What is software scope?

Software scope is a well-defined boundary, which encompasses all the activities that are done to develop and deliver the software product.

Q) What are function points?

Function points are the various features provided by the software product. It is considered as a unit of measurement for software size.

Q) What is baseline?

Baseline is a measurement that defines completeness of a phase. After all activities associated with a phase are accomplished, the phase is complete and acts as a baseline for next phase.

Q) What are software requirements?

Software requirements are functional description of proposed software system. Requirements are assumed to be the description of target system, its functionalities and features.

Q) What are functional requirements?

Functional requirements are functional features expected by users from the proposed software product.

Q) What are non-functional requirements?

Non-functional requirements are implicit and are related to security, performance, look and feel of user interface, cost etc.

Q) What is software measure?

Software Measures can be understood as a process of quantifying and symbolizing various attributes and aspects of software.

Q) What is software metric?

Software Metrics provide measures for various aspects of software process and software product. They are divided into –

* Requirement metrics: Length requirements, completeness
* Product metrics: Lines of Code, Object oriented metrics, design and test metrics
* Process metrics: Evaluate and track budget, schedule, human resource.

Q) What is modularization?

Modularization is a technique to divide a software system into multiple discreet modules, which are expected to carry out task(s) independently.

Q) What is concurrency and how it is achieved in software?

Concurrency is the tendency of events or actions to happen simultaneously. In software, when two or more processes execute simultaneously, they are called concurrent processes.

Q) What is cohesion?

Cohesion measures, the degree of intra-dependability among the elements of the module.

Q) What is coupling?

Coupling measures, the level of inter-dependability among modules of a program.

Q) What is the difference between structured English and Pseudo Code?

Structured English is native English language used to write the structure of a program module by using programming language keywords, whereas, Pseudo Code is closer to programming language and uses native English language words or sentences to write parts of code.

Q) What is structured design?

Structured design is a conceptualization of problem into several well-organized elements of solution. It is concerned with the solution design and based on ‘divide and conquer’ strategy.

Q) What is the difference between function oriented and object-oriented design?

Function-oriented design is comprised of many smaller sub-systems known as functions. Each function can perform significant task in the system. Object oriented design works around the real-world objects (entities), their classes (categories) and methods operating on objects (functions).

Q) Briefly define top-down and bottom-up design model.

Top-down model starts with generalized view of system and decomposes it to more specific ones, whereas bottom-up model starts with most specific and basic components first and keeps composing the components to get higher level of abstraction.

Q) What is functional programming?

Functional programming is style of programming language, which uses the concepts of mathematical function. It provides means of computation as mathematical functions, which produces results irrespective of program state.

Q) Differentiate validation and verification?

Validation checks if the product is made as per user requirements whereas verification checks if proper steps are followed to develop the product.

Validation confirms the right product and verification confirms if the product is built in a right way.

Q) What is black-box and white-box testing?

Black box testing checks if the desired outputs are produced when valid input values are given. It does not verify the actual implementation of the program.

White box testing not only checks for desired and valid output when valid input is provided but also it checks if the code is implemented correctly.

Q) Explain Big O-Notation in simple terms.

Big-O notation is used in computer science to describe the performance or complexity of an algorithm. It describes how the runtime or space requirement of a function grows as the input grows.

Q) How would you explain APIs to non-technical stakeholders?

An API tells programmers how to automate a product—anything from web apps like Twitter all the way down to Windows itself. For example, I could use Twitter’s API to fetch our company’s most recent tweets and then display them on our website. That way our social media specialist can simply tweet, without having to always take extra steps to copy the tweet to our website.

What is Dynamic Programming? And DP problems?

Dynamic Programming is the paradigm that solves a complex problem by breaking it into subproblems and stores the results of subproblems to avoid computing the same result again.

* Overlapping Subproblems: Dynamic programming is used when solutions of some subproblems are needed again and again.
* Optimal Substructure: A problem has optimal substructure property if optimal solution of the given problem can be obtained using optimal solutions of its subproblems.

Steps to solve any Dynamic programming problem:

* Decide a state expression with least parameters
* Formulate state relationship
* Do tabulation/memorization to avoid recomputing previously computed values.

What is an algorithm?

An algorithm is thus a sequence of computational steps that transform the input into the output.

What is a Data Structure?

A data structure is a way of organizing the data so that the data can be used efficiently.

What are linear and nonlinear data Structures?

* Linear: A data structure is said to be linear if its elements form a sequence or a linear list. Examples: Array. Linked List, Stacks and Queues
* Non-Linear: A data structure is said to be non-linear if traversal of nodes is nonlinear in nature. Example: Graph and Trees.

Can Binary Search be used for linked lists?

Since random access is not allowed in linked list, we cannot reach the middle element in O (1) time. Therefore, Binary Search is not possible for linked lists.

Common time complexities:

* Binary Search: O (log n)
* Quicksort: O (n log n)
* Merge Sort: O (n log n)
* Heap Sort: O (n log n)
* Bubble Sort: O (n^2)
* Insertion Sort: O (n^2)
* Selection Sort: O (n^2)

Common Data Structures and their complexities:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Structure | Access | Search | Insertion | Deletion |
| Array | O (1) | O (n) | O (n) | O (n) |
| LinkedList | O (n) | O (n) | O (1) | O (1) |
| B-Tree | O (log n) | O (log n) | O (log n) | O (log n) |
| Stack | O (n) | O (n) | O (1) | O (1) |
| Queue | O (n) | O (n) | O (1) | O (1) |
| Hash Table | N/A | O (1) | O (1) | O (1) |
| Binary Search Tree | O (log n) | O (log n) | O (log n) | O (log n) |

Stacks vs Queues:

* Stack is a linear data structure which follows the LIFO (Last in First Out) order. Basic operations of stack are: Push, Pop, Peek.
* Queue is a linear structure which follows the First in First Out (FIFO) order. Basic operations on queue: Enqueue, Dequeue, Front, Rear.

What is Infix, prefix, Postfix notations?

* Infix notation: X + Y – Operators are written in-between their operands. This is the usual way we write expressions. An expression such as => A \* (B + C) / D
* Postfix notation (Reverse Polish notation): X Y + Operators are written after their operands. The postfix expression given above is equivalent to => A B C + \* D/
* Prefix notation (Polish notation): + X Y Operators are written before their operands. The expressions given above are equivalent to => / \* A + B C D

What is a Linked List and What are its types?

A linked list is a linear data structure where each element is a separate object. Each element (that is node) of a list is comprising of two items – the data and a reference to the next node.

* Singly Linked List: Every node stores address or reference of next node in list and the last node has next reference as NULL. For example, 1->2->3->4->NULL
* Doubly Linked List: Here, here are two references associated with each node, One of the reference points to the next node and one to the previous node. E.g. NULL<-1<->2<->3->NULL
* Circular Linked List: Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list. E.g. 1->2->3->1 [The next pointer of last node is pointing to the first]

How to implement a stack using queue?

A stack can be implemented using two queues

How to implement a queue using stack?

A queue can be implemented using two stacks

Which Data Structures should be used to implement LRU cache?

* Queue which is implemented using a doubly linked list. The maximum size of the queue will be equal to the total number of frames available (cache size). The most recently used pages will be near rear end and least recently pages will be near front end.
* A Hash with page number as key and address of the corresponding queue node as value.

How to check if a given Binary Tree is BST or not?

If in order traversal of a binary tree is sorted, then the binary tree is BST. The idea is to simply do in order traversal and while traversing keep track of previous key value. If current key value is greater, then continue, else return false.