

1. OOP

Class is a user-defined data type. Class defines the type definition of category of things. It defines a data type, but it does not define the data it just specifies the structure of data. You can create N number of objects from a class.

Object is the instance of a class. A class provides a blueprint for objects. So you can create an object from a class. The objects of a class are declared with the same sort of declaration that we declare variables of basic types.

OOP is a methodology or paradigm that provides many concepts. The basic concepts of Object Oriented Programming are given below:

Encapsulation: Encapsulation is a mechanism which binds the data and associated operations together and thus hide the data from outside world. Encapsulation is also known as data hiding.

Private: The data members and functions cannot be accessed from outside the class.

Protected: The data members and functions are accessible to derived class only.

Public: The data members and functions can be accessed from outside the class.

Abstraction: Abstraction is used to hide the internal implementations and show only the necessary details to the outer world. Data abstraction is implemented using interfaces and abstract classes in C++.

Abstract class: A class that is declared as abstract is known as abstract class. It needs to be extended and its method implemented. It cannot be instantiated. A method that is declared as abstract and does not have implementation is known as abstract method.

Interface: It has static constants and abstract methods only. It is a mechanism to achieve fully abstraction. There can be only abstract methods in the java interface not method body. It is used to achieve fully abstraction and multiple inheritance in Java. It cannot be instantiated just like abstract class.

Inheritance: Inheritance is used to inherit the property of one class into another class. It facilitates you to define one class in term of another class. Inheritance is a mechanism in which one object acquires all the properties and behavior of another object of another class. It represents IS-A relationship. It is used for Code Reusability and Method Overriding.

Polymorphism is the ability of an object to take on many forms. The most common use of **polymorphism** in OOP occurs when a parent class reference is used to refer to a child class object. Any object that can pass more than one IS-A test is considered to be polymorphic.

Overloading: (Compile Time/ static) C++ facilitates you to specify more than one definition for a function name or an operator in the same scope. Parameter must be different. It is called function overloading and operator overloading respectively.

Overriding: (Runtime/Dynamic) If you inherit a class into a derived class and provide a definition for one of the base class's function again inside the derived class, Parameter must be same, then this function is called overridden function and this mechanism is known as function overriding.

Aggregation and Composition: Aggregation represents weak relationship whereas composition represents strong relationship. For example: bike has an indicator (aggregation) but bike has an engine (composition).

Array Vs List: Array is a collection of homogeneous elements while list is a collection of heterogeneous elements.

Array memory allocation is static and continuous while List memory allocation is dynamic and random.

In Array, users don't need to keep in track of next memory allocation while in list user has to keep in track of next location where memory is allocated.

Static Variable is used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc. Static variable gets memory only once in class area at the time of class loading.

Static Method belongs to the class rather than object of a class. A static method can be invoked without the need for creating an instance of a class. Static method can access static data member and can change the value of it.

2. DSA

	Time				
Sort	Average	Best	Worst	Space	Stability
Bubble sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Constant	Stable
Modified Bubble sort	$O(n^2)$	$O(n)$	$O(n^2)$	Constant	Stable
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Constant	Stable
Insertion Sort	$O(n^2)$	$O(n)$	$O(n^2)$	Constant	Stable
Heap Sort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	Constant	Instable
Merge Sort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$	Depends	Stable
Quicksort	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$	Constant	Stable

RDBMS uses Array data structure

Network data model uses Graph

Hierarchal data model uses Trees

Binary trees: A Binary Tree is a type of data structure that has two nodes: A left node and a right node. In programming, binary trees are actually an extension of the linked list structures.

Linked list: it is a sequence of nodes in which each node is connected to the node following it. It makes a chain like link of data storage.

Priority Queue: Two queues are needed. One queue is used for actual storing of data and another for storing priorities.

Stack: it is used to perform recursion because of its LIFO (Last in First Out) property. It knows whom to return when the function has to return. A stack is a data structure in which only the top element can be accessed.

Merge sort: *it takes advantage of the ease of merging already sorted lists into a new sorted list. It starts by comparing every two elements (i.e., 1 with 2, then 3 with 4...) and swapping them if the first should come after the second. It then merges each of the resulting lists of two into lists of four, then merges those lists of four, and so on; until at last two lists are merged into the final sorted list. Of the algorithms described here, this is the first that scales well to very large lists, because its worst-case running time is $O(n \log n)$.*

Quicksort is a divide and conquer algorithm which relies on a partition operation: to partition an array an element called a pivot is selected. All elements smaller than the pivot are moved before it and all greater elements are moved after it. This can be done efficiently in linear time and in-place. The lesser and greater sub lists are then recursively sorted.

B-tree is a self-balancing tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree is a generalization of a binary search tree in that a node can have more than two children. Unlike self-balancing binary search trees, the B-tree is optimized for systems that read and write large blocks of data. B-trees are a good example of a data structure for external memory. It is commonly used in databases and file systems.

Heap is a specialized tree-based data structure that satisfies the **heap** property: If A is a parent node of B then the key (the value) of node A is ordered with respect to the key of node B with the same ordering applying across the **heap**

Hash table (hash map) is a data structure used to implement an associative array, a structure that can **map** keys to values. A **hash table** uses a **hash** function to compute an index into an array of buckets or slots, from which the desired value can be found.

3. OS

Operating System: The operating system is a software program that facilitates computer hardware to communicate and operate with the computer software. It is the most important part of a computer system without it computer is just like a box.

Operating System Purpose: i) It is designed to make sure that a computer system performs well by managing its computational activities. ii) It provides an environment for the development and execution of programs.

Operating Systems exm: i) distributed operating systems, ii) Multi-programmed operating systems, iii) Real-time operating systems

Multiprocessor system: As many as processors are increased, you will get the considerable increment in throughput. It is cost effective also because they can share resources. So, the overall reliability increases.

Virtual Memory: Virtual memory is a very useful memory management technique which enables processes to execute outside of memory. This technique is especially used when an executing program cannot fit in the physical memory.

Thread: A thread is a basic unit of CPU utilization. It consists of a thread ID, program counter, register set and a stack.

FCFS: FCFS stands for First Come, First Served. It is a type of scheduling algorithm. In this scheme, if a process requests the CPU first, it is allocated to the CPU first. Its implementation is managed by a FIFO queue.

4. Database

Database: Database is a logical, consistent and organized collection of data that it can easily be accessed, managed and updated.

Primary key is a table column (or combination of columns) designated to uniquely identify all table records. A **primary key's** main features are: It must contain a unique value for each row of data. It cannot contain null values.

Foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. In simpler words, the **foreign key** is defined in a second table, but it refers to the **primary key** in the first table.

Normalization: Normalization is a process of analyzing the given relation schemas according to their functional dependencies. It is used to minimize redundancy and also minimize insertion, deletion and update distractions

E-R model: E-R model is a short name for Entity Relationship model. This model is based on real world. It contains basic objects (known as entities) and relationship among these objects.

Entity: Entity is a set of attributes in a database.

Stored Procedure: A stored procedure is a named group of SQL statements that have been previously created and stored in the server database.

5. Software Engineering

Software Engineering is an engineering branch associated with software system development.

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

SDLC Models: Waterfall Model, Iterative Model, Spiral model, V-model, Incremental model

Phases of SDLC: The generic phases of SDLC are: Requirement Gathering, System Analysis and Design, Coding, Testing and implementation. The phases depend upon the model we choose to develop software.

Process is a repetitive collection of interrelated tasks aimed at achieving a certain goal. **Project** is a unique endeavor with a beginning and an end undertaken to achieve a goal. **Product** is *what* we're actually building.

Waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases, typically:

Requirements analysis , Software design, Implementation, Testing, Integration, , Deployment , Maintenance

Incremental model:

A series of mini-Waterfalls are performed, where all phases of the Waterfall are completed for a small part of a system, before proceeding to the next increment, or

Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of a system, or

The initial software concept, requirements analysis, and design of architecture and system core are defined via Waterfall, followed by incremental implementation, which culminates in installing the final version, a working system.

Software Project Management: Software project management is process of managing all activities like time, cost and quality management involved in software development.

Software Project Manager: A software project manager is a person who undertakes the responsibility of carrying out the software project.

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. Requirements convey the expectations of users from the system.

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

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

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.

Black-box 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.

6. Networking

Open Systems Interconnection model (OSI model) is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system. It has 7 layers.

1. Application layer: Application layer defines TCP/IP application protocols and how host programs interface with Transport layer services to use the network.

2. Presentation layer: The purpose of Transport layer is to permit devices on the source and destination hosts to carry on a conversation. Transport layer defines the level of service and status of the connection used when transporting data.

3. Session layer: Internet layer pack data into data packets known as IP datagrams, which contain source and destination address (logical address or IP address) information that is used to forward the datagrams between hosts and across networks.

4. Transport layer: it is responsible for end-to-end communication over a network. It provides logical communication between application processes running on different hosts within a layered architecture of protocols and other network components.

5. Network layer: Network Access Layer defines details of how data is physically sent through the network, including how bits are electrically or optically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber.

6. Data link layer: is the protocol **layer** that transfers **data** between adjacent network nodes in a wide area network (WAN) or between nodes on the same local area network (LAN) segment.

7. Physical layer: deals with the **physical** connectivity of two different stations.

This **layer** defines the hardware equipment, cabling, wiring, frequencies, pulses used to represent binary signals

TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules defining the Internet.

IP address consists of two **parts**, one identifying the network and one identifying the node, or host. The Class of the **address** determines which **part** belongs to the network **address** and which **part** belongs to the node **address**. All nodes on a given network share the same network prefix but must have a unique host number.

Difference between TCP and UDP

TCP	UDP
<i>Reliability:</i> TCP is connection-oriented protocol. When a message send it will get delivered unless connections fails. If connection lost, the server will request the lost part. There is no corruption while transferring a message.	<i>Reliability:</i> UDP is connectionless protocol. When you a send a data or message, you don't know if it'll get there, it could get lost on the way. There may be corruption while transferring a message.
<i>Ordered:</i> first message will get there first.	<i>Ordered:</i> no ordered
<i>Heavyweight:</i> –	<i>Lightweight:</i> It's just fire and forget! This means it's a lot quicker, and the network card / OS have to do very little work to translate the data back from the packets.
<i>Examples:</i> World Wide, e-mail	<i>Examples:</i> Domain Name System (DNS UDP port 53), streaming media movies,

DNS: The Domain Name System (**DNS**) is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities