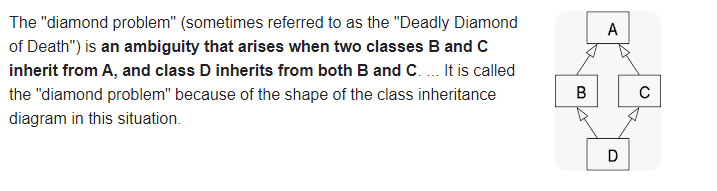
**OOP**

Q1: What is Polymorphism  
Ans: Abstract class (Pure Virtual Fuction), overrideing,

Q2: Singelton class?  
Ans: Which can have only one object.   
Q3: Difference between Overriding(dynamic) and overloading(Static polymorphism)?  
Q3: where to use Pure virtual function and where to use ploymophism?  
Ans:  
Q4 :What is Diamond problem in OOP?

  
Q5 :Inheritance ?

Ans: a>b>c Construstor a,b,c Distructor c,b,a?

Q6: what does Virtual function do ?  
Ans: Late Binding and Prevent early binding. Virtual function is like abstract function and its functionality can be over ridden in its child class. It is implemented by using virtual.

Q6: Deep Copy , Shellow Copy?  
Ans: A(b) A->B in shallow copy both objects points same Address .In Deep both have their own memories.

String::String(const char \*str) { size = strlen(str); s = new char[size+1]; strcpy(s, str); }

Q7: Aggregation, Composition

Aggregation defines a one-way relationship that specifies a 'has-a' relationship between two classes.

Composition is a special type of Aggregation.

For example: **A car has a engine, a window has a button, a zoo has a tiger**. Composition is a special case of aggregation. In other words, a restricted aggregation is called composition. When an object contains the other object and the contained object cannot exist without the other object, then it is called composition.

Q8: Type of inheritance…

* Single inheritance. (parent have inly single child class)
* Multi-level inheritance. (A->B->C)
* Multiple inheritance. (A & B are parent -> C is child)
* Multipath inheritance.
* Hierarchical Inheritance.
* Hybrid Inheritance.

Q9: Structure and Class difference?  
Ans: Structure by default public, no constructors/destructor, no inheritance,

Class by default private, Yes Constructor

**Data Structure**

# Q1: Delete center node in single or double link list? Q2: how to break loop of cirlcular link list? Q3: Reverse a link list using single loop? Q4: Hashing ?

# is a technique of mapping a large chunk of data into small tables using a hashing function.

# Q5: Complexity of sortings?

# 

# Q6: Trees ? Binary tree trivesals infix , post fix, refix? Code? left have small values and right large… A tree is a hierarchical data structure defined as a collection of nodes. Nodes represent value and nodes are connected by edges. A tree has the following properties: The tree has one node called root

# Q7: minimum spaning tree?

# undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible. Ans: Kruskal’s Algorithm and prims algo

# Q8:why we use AVL tree? Ans: minimum height of tree adjust krta ha …..

**DataBases**

Q1: Normalization?

Normalization is the process of organizing data in a database.

## **1NF (First Normal Form) Rules**

* Each table cell should contain a single value.
* Each record needs to be unique.

## 2NF (Second Normal Form) Rules

* Rule 1- Be in 1NF
* Rule 2- Single Column Primary Key that does not functionally dependant on any subset of candidate key relation

## 3NF (Third Normal Form) Rules

* Rule 1- Be in 2NF
* Rule 2- Has no transitive functional dependencies

Q2: How to resolve many to many relation? ()  
Q3: Difference btw SP and function?  
Q4:Indexing ? best for searching..  
Q5: Joins?  
Q6: Second highest number in DB?

**Programming Logics**

Q1: Find K ki power N using recursion?  
Ans: int calculatePower(int base, int powerRaised)

{

if (powerRaised != 0)

return (base\*calculatePower(base, powerRaised-1));

else

return 1;

}

Q2: Reverse an array using one loop (N times)?  
Ans:  
Q3: Binary to Decimal 0.2 to 2?  
Ans: int convertBinaryToDecimal(long long n)

{

int decimalNumber = 0, i = 0, remainder;

while (n!=0)

{

remainder = n%10;

n /= 10;

decimalNumber += remainder\*pow(2,i);

++i;

}

return decimalNumber;

}

Q4: Swap to values without using third variable?  
Q5: Dynamic 2D array declaration?  
Ans: int\*\* a = new int\*[rowCount];

for(int i = 0; i < rowCount; ++i)

a[i] = new int[colCount];

Q4: Display array using Pointer?

int a[5]={1,2,3,4};

for(int i=0;i<4;i++)

{ cout<<\*(a+i); }

**Q5 : what is pointer to pointer?  
int \*a= &b; (1d array)**

**Int \*\*ptr=&a; (2d array)**

**Int \*\*\*ptr2=&ptr; (3d array)**

**Q5: 2d array using ptr?  
Ans: \*(\*(a+i)+j)**

**a[i][j]**

**Q6:** structure of classes?