

# Types of Programming

## Imperative

```
num = input()
if num % 2 == 0:
    print("Even
Number")
else:
    print("odd number")
```

## Functional

```
def evenodd(n):
    if n%2 == 0:
        return "Even"
    else:
        return "odd"
print(evenodd(56))
print(evenodd(90))
```

## Object Oriented

```
class checker:
    def __init__(self, n):
        if n%2 == 0:
            print("even")
        else:
            print("odd")
num1 = checker(34)
num2 = checker(12)
```

Imperative m ek baar mei ek no. k uppr work krte hai

Functional m ek function create krke usme parameter dete hai , isme multiple elements ko ek sath check kr skte hai

Object m thoda bht function ki trf hota pr isme aati hai classes

## What is special about OOPS ?

- ④ Make your code more reusable.
- ④ Easier to work with large programs.
- ④ OOP programs prevent you from repeating code .
- ④ OOP provide you security.

**Car Factory**





**Ferrari**

- ⊕ Body Types      Covered
- ⊕ Tyres              2 Layered
- ⊕ Engine Type      8 Cycle



**Lord Alto**

- ⊕ Body Types      Covered
- ⊕ Tyres              1 Layered
- ⊕ Engine Type      2 Cycle



**Sigma splendor**

- ⊕ Body Types      Open
- ⊕ Tyres              3 Layered
- ⊕ Engine Type      8 Cycle

## what are Objects ?

objects are instances of classes.

- Matlab seedhi bhasha mai ek class ka use krke uske bacche bnana.

**Example:**

In the last example alto , bike and ferrari are objects.

## what is Encapsulation ?

- ④ In programming, encapsulation is about keeping some information (data) safe and only letting it be changed or looked at in specific ways.
- ④ Matlab safe rakhna apne data ko aur koi bhi bahar se access na kar sake apni class ke andar.

## Example :

- ④ Like in this image a class is encapsulated and only people who have paid the class fees can attend the class no one else can NOO ONE !!!!!



# What is Polymorphism ?

**Polymorphism means having many forms.**

- matlab eak he cheez diffrent tasks perform kar paa raha hai.

## Example :

- Here the phone have both the qualities of a camera and a calculator so it is showing a polymorphic bheaviour

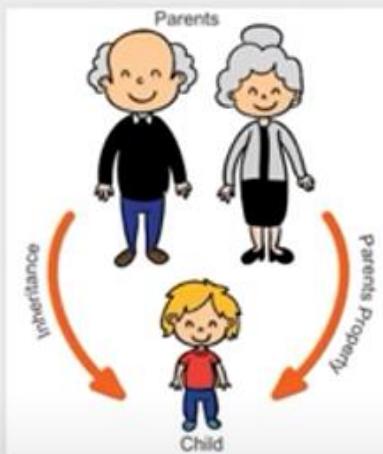


# What is Inheritance ?

**when one class inherits some feature from another class this phenomena is known as inheritance**

- **eak class jab doosri class se cheeze transfer karti hai usko inheritance kahete hai.**

## Example :



- A child inherits the properties of their parents  
this is known as inheritance

## What is Abstraction ?

When we only see the essential part of our code and hides the rest is the process of Abstraction.

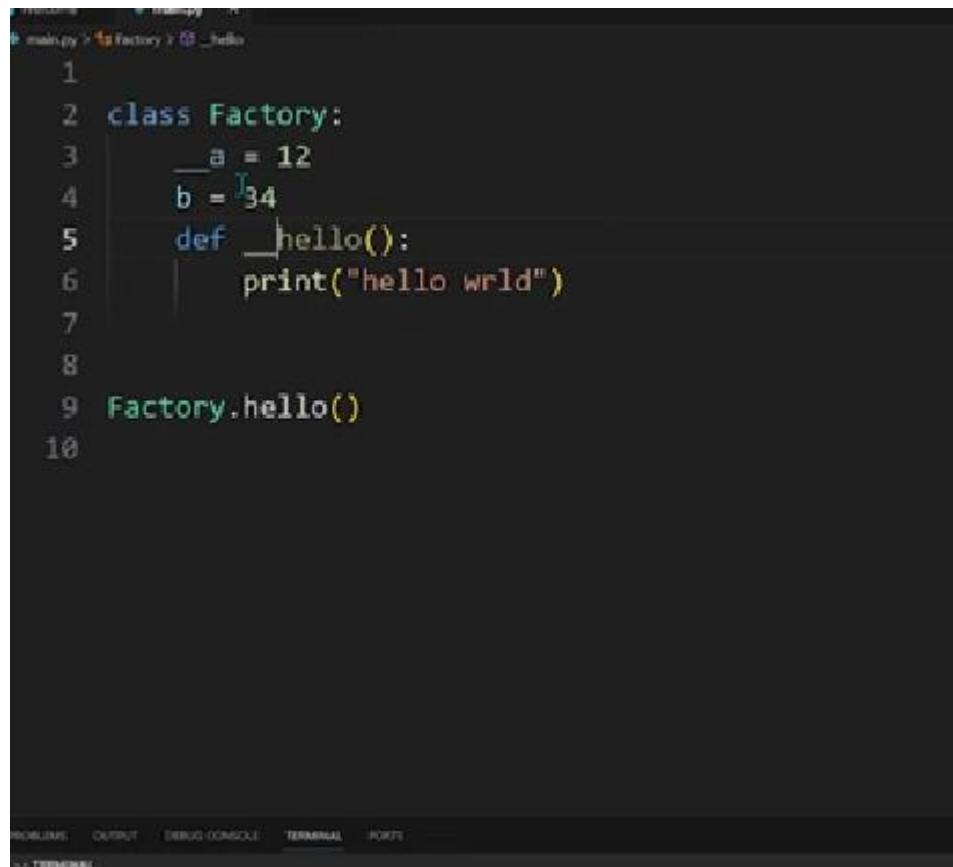


Plug board



shipping box

- matlab sirf kaam ki cheez ko dikhana aur sab ko chupa lena .



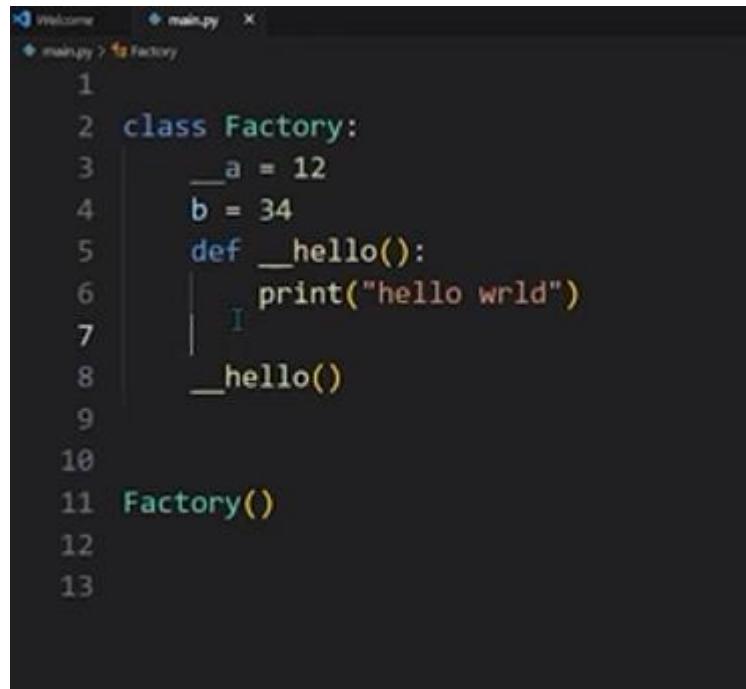
```

1
2 class Factory:
3     __a = 12
4     b = 34
5     def __hello():
6         print("hello wrld")
7
8
9 Factory.hello()
10

```

double underscore

for making private (encapsulated)



```

1
2 class Factory:
3     __a = 12
4     b = 34
5     def __hello():
6         print("hello wrld")
7
8     _hello()
9
10
11 Factory()
12
13

```

class k andr private access kr skte ho 2 underscore

```
+ main.py > ..
1
2 class Factory:
3     a = 12
4     b = 34
5     @classmethod
6     def hello():
7         print("yoo")
8
9
10 Factory()
11
```

decorator is used , to ye bhi class

private hogi to bahr se access nah ikr skte function

Attribute – inside class / variable – outside class

Method -inside / function -outside

```
+ main.py
class Factory:
    a = 12
    b = 13

obj = Factory()
```

obj variable ek object bn gya , object k paas saari class ki powers hoti hai

Object hua simply ek variable jisme humne class ko call krdiya

```

1
2 class Factory(bodytype,tyres)
3
4

```

jesse function hum parameters lete hai wese class m agr lena ho to ese direct nahi le skte isliye hum constructor banate hai

```

1
2 class Fa (method) def __init__(self: Self@Factory) -> None
3     def __init__(self):
4         print("hello world")
5
6
7 Factory()
8

```

Function call nahi kra pr ye automatic call hojayega aur hello world print hojaayega

Constructor is an initialization function

```

1
2 class Factory:
3     def __init__(self,BT,tyres,ET):
4         print("hello world")
5
6
7
8 Factory()
9

```

to agr humein class k andr kuchh parameters chahiye to hum constructor yaa initialization function k andr vo parameters dete hai

```

class Factory:
    def __init__(self,BT,tyres,ET):
        print(BT,tyres,ET)

ferrari = Factory("covered",4,"8 cycle")

```

self call krta hai object ko ,

yaha Ferrari ek object , self m Ferrari ki location hai

To mtlb jb bhi hum kisi object to uss object ki location kaha jaakr store hoti hai , self mei

```

1
2 class Factory:
3     def __init__(self,BT,tyres,ET):
4         self.BT = BT
5         self.tyres = tyres
6         self.ET = ET
7
8
9
10 ferrari = Factory("covered",4,"8 cycle")
11
12 print(ferrari.BT)
13

```

ese location ki jgh ab value

print hogi

print(Ferrari) m object ki location print hoti hai

```

class Factory:
    def __init__(self,BT,tyres,ET):
        self.BT = BT
        self.tyres = tyres
        self.ET = ET

ferrari = Factory("covered",4,"8 cycle")
alto = Factory("covered",4,"4 cycle")
splendor = Factory("open",2,"1 cycle")

print(alto.tyres)

```

self bolta mereko aap object to uski

location mai rkhunga

Agr hum kahi p self ka use kr rhe hai jese uppr BT to vo kya hogya object attribute jese a=2 k hota hai class attribute

```

8     def hello():
9         print("hello brother")
10
11
12
13 ferrari = Factory("covered",4,"8 cycle")
14 alto = Factory("covered",4,"4 cycle")
15 splendor = Factory("open",2,"1 cycle")
16
17
18 splendor.hello()

```

jab bhi kisi function ko aap ek object through call kroge to vo object ek argument ki trf chla jaata hai

To direct call m phir vo error deta hai

```

def hello(self):
    print("hello brother")

ferrari = Factory("covered")
alto = Factory("covered",4,"4 cycle")
splendor = Factory("open",2,"1 cycle")

splendor.hello()

```

ab humne self likhdiya to argument error nahi jaayega , ye hai object method

```

5         self.tyres = tyres
6         self.ET = ET
7
8     def showdetails(self):
9         print(f"your details are\n {self.BT},{self.tyres},{self.ET}")
10
11
12
13 ferrari = Factory("covered",4,"8 cycle")
14 alto = Factory("covered",4,"4 cycle")
15 splendor = Factory("open",2,"1 cycle")
16
17
18 splendor.showdetails()

```

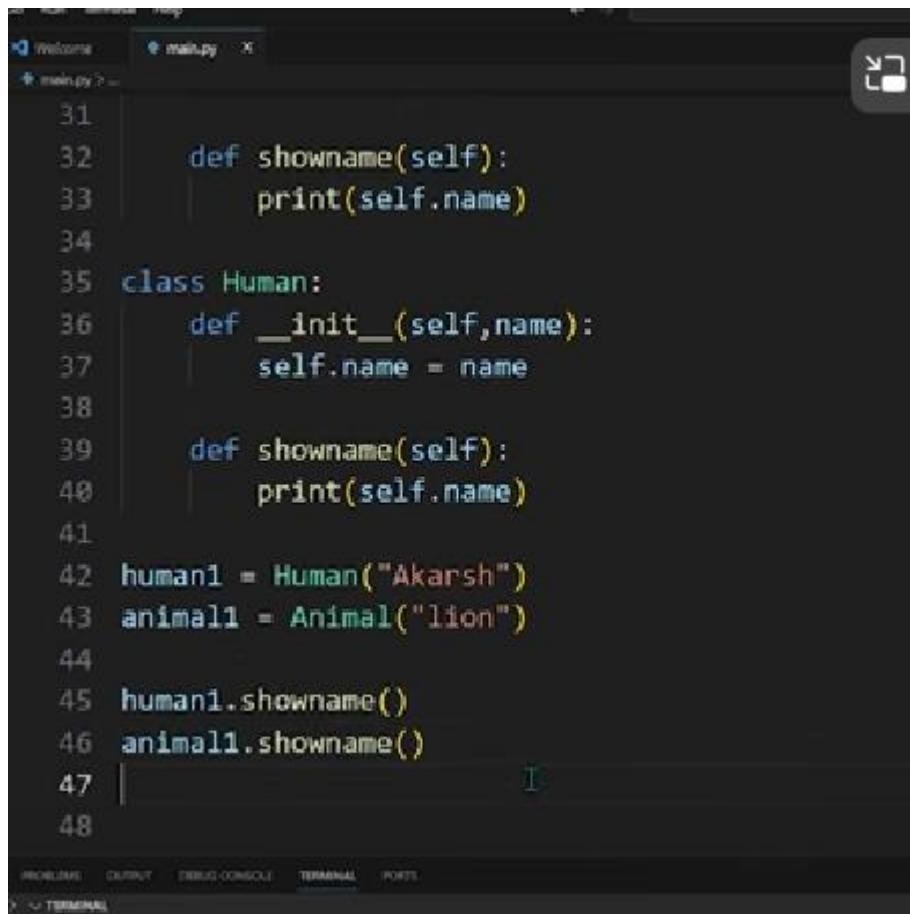
```
class Honda(Factory):
    pass
    # inheritance , factory is the super class
```

```
Welcome + main.py x
mainpy >= 1
2     self.tyres = tyres
3     self.ET = ET
4
5     def showdetails(self):
6         print(f"your details are\n {self.BT},{self.tyres},{self.ET}")
7
8 class Honda(Factory):
9     def __init__(self, BT, tyres, ET,glass):
10        super().__init__(BT, tyres, ET)
11        self.glass = glass
12
13 obj1 = Honda(1,2,3,4)
14
15 obj1.showdetails()
```

agr sub class m ek naya parameter add krna hua

**Multiple inheritance – jb mai 2 class ko inherit krta hu ek single class k andr**

**Multilevel inheritance – ek class uss class ko inherit kr rhi hai jo class already kis aur class ko bhi kr rhi hai means , sub class of the sub class**



A screenshot of a Python code editor showing a file named 'main.py'. The code defines two classes, 'Human' and 'Animal', both with a 'showname' method that prints the object's name. An instance of 'Human' is created with name 'Akarsh', and an instance of 'Animal' is created with name 'lion'. Both objects are then called with their 'showname' method. The code editor interface includes tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS', along with a terminal window at the bottom.

```
31
32     def showname(self):
33         print(self.name)
34
35 class Human:
36     def __init__(self, name):
37         self.name = name
38
39     def showname(self):
40         print(self.name)
41
42 human1 = Human("Akarsh")
43 animal1 = Animal("lion")
44
45 human1.showname()
46 animal1.showname()
47
48
```

polymorphism done

class m same function hai pr object alg alg hai

```

27
28 class Animal:
29     def __init__(self, name):
30         self.name = name
31
32     def showname(self):
33         print(self.name)
34
35 class Human(Animal):
36     def showname(self):
37         super().showname()
38         print("asfsf" + self.name)
39
40
41 obj1 = Human("akarsh")
42
43 obj1.showname()
44

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

> TERMINAL

```

PS C:\Users\Akarsh\Desktop\OOPS> & C:/Users/Akarsh/AppData/Local/Programs/Python/3.8/Akarsh
PS C:\Users\Akarsh\Desktop\OOPS> & C:/Users/Akarsh/AppData/Local/Programs/Python/3.8/Akarsh
15m
PS C:\Users\Akarsh\Desktop\OOPS> & C:/Users/Akarsh/AppData/Local/Programs/Python/3.8/Akarsh
asfsf akarsh
PS C:\Users\Akarsh\Desktop\OOPS> & C:/Users/Akarsh/AppData/Local/Programs/Python/3.8/Akarsh
akarsh
asfsf akarsh
PS C:\Users\Akarsh\Desktop\OOPS>

```

**method overriding**

Python m abstraction exist hi nahi krta

from abc import ABC , abstractmethod

to hum ye library use krte hai

```

8 from abc import ABC , abstractmethod
9
10 class Shape(ABC):
11     @abstractmethod
12         def area():
13             pass
14
15     @abstractmethod
16         def perimeter():
17             pass
18
19 class Circle(Shape):
20     def __init__(self, radius):
21         self.radius = radius
22
23
24 cir1 = Circle(12)

```

yaha hum normal class se  
abstract class ko lene ki koshish kr rhe hai pr yaha ek error aajayega ki hum esa nah ikr skte aur uske  
liye phir humein

```

39 class Circle(Shape):
40     def __init__(self, radius):
41         self.radius = radius
42
43     def area():
44         pass
45
46     def perimeter():
47         pass
48
49
50 cir1 = Circle(12)

```

ye 2 abstract methods banani

padegi

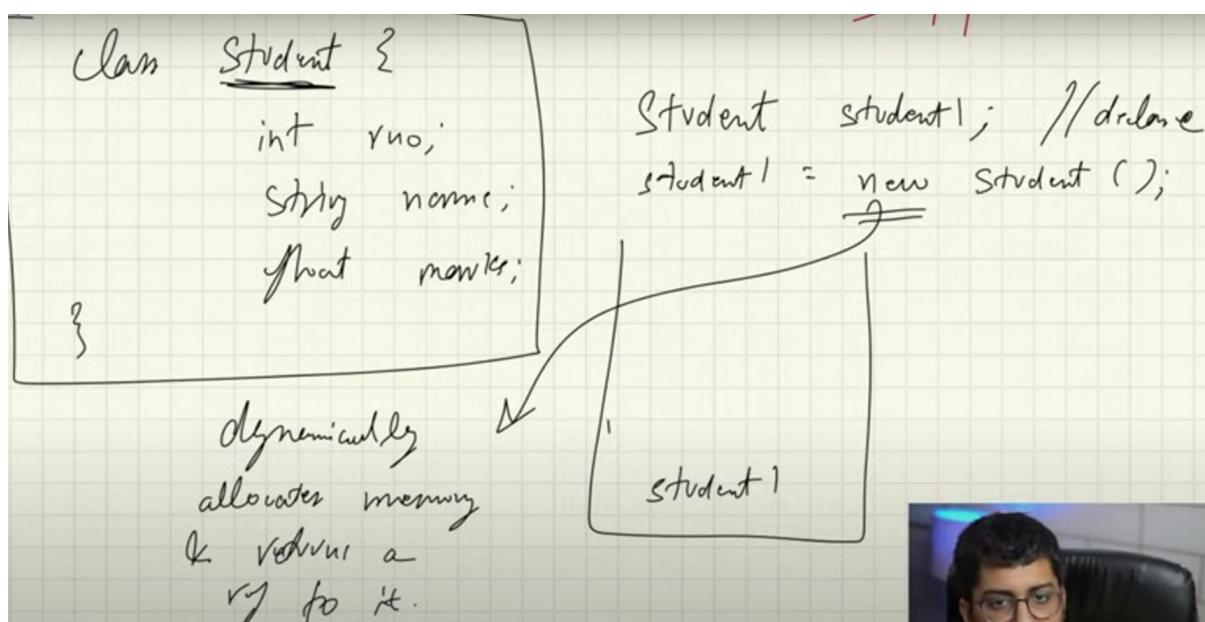
*Class → logical construct*

*Object → physical reality // Occupies space in memory -*

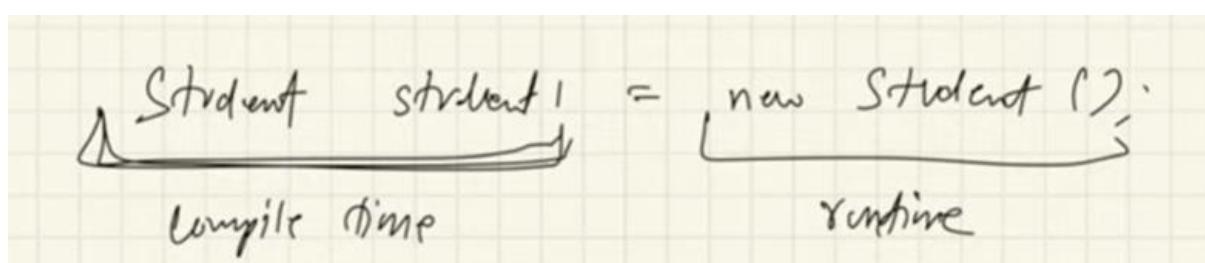
### 3 main properties of object :

1. State
2. Identity
3. behaviour

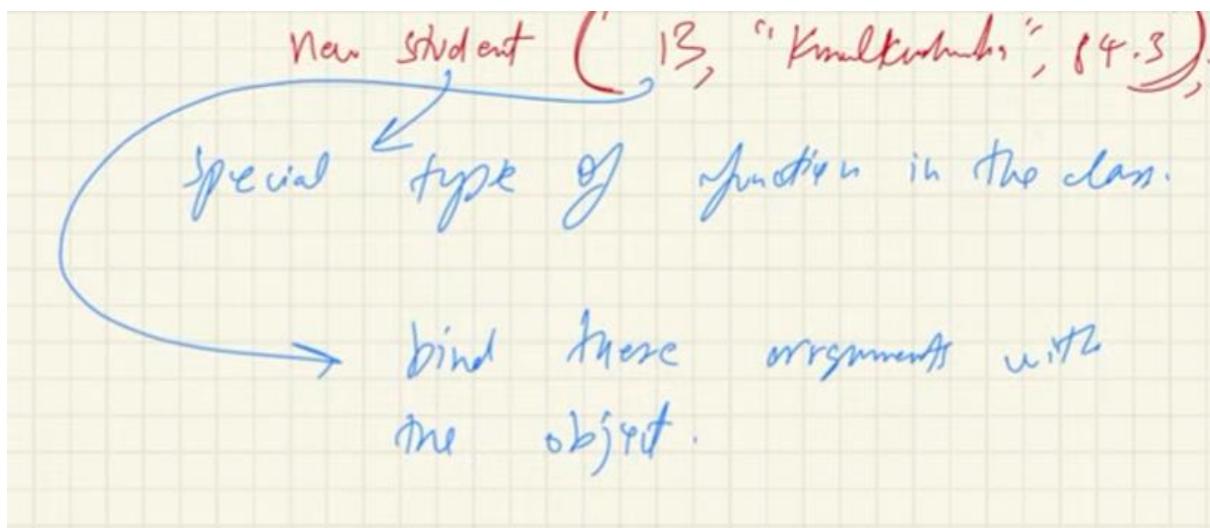
**dot operator** links the instance variable to its reference variable



**new function to allocate**



Constructor is special function, that runs when you create an object and it allocates some variables.



```
Student () {
    kunal.rno = 13;
    kunal.name = "Kunal Kushwaha";
    kunal.marks = 88.5f;
}
```

```
Student () {
    this.rno = 13;
    this.name = "Kunal Kushwaha";
    this.marks = 88.5f;
}
```

Now it can take any name we say on place of this  
 when new object is created it goes inside the constructor

So basically it is a one single word to access all the objects

```
Student kunal = new Student();
Student rahul = new Student();
kunal.rno = 13; like this

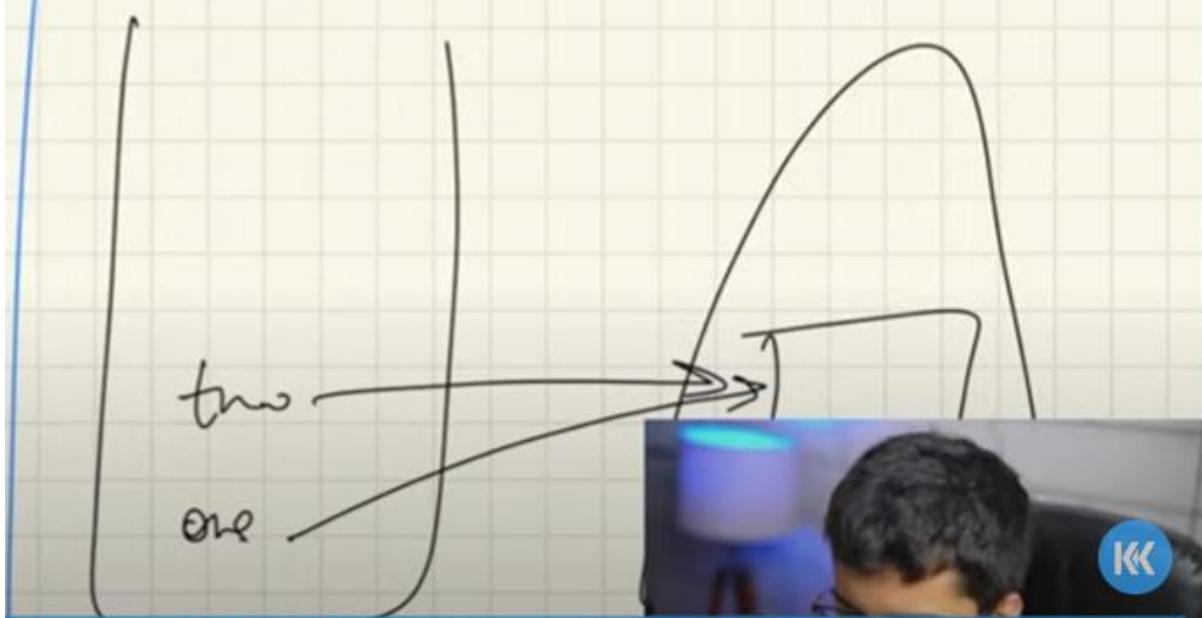
void greeting() {
    System.out.println("Hello! My name is " + this.name);
}
```

use

(there are no primitives in python) in java primitives are stored in stack memory

Student one = new Student();

Student two = one;



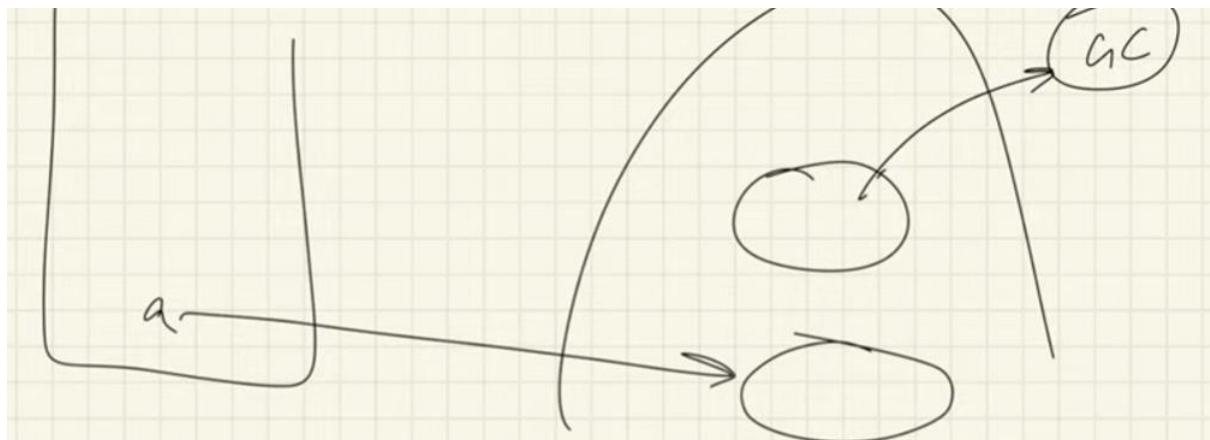
Wrapper classes – creating using new

used to convert primitive data types into objects

The final keyword in Java is a non-access modifier used to restrict the modification of variables, methods, and classes. It ensures that certain aspects of a program remain constant and unchangeable

final int INCREASE = 2;

always initialize while declaring.



finalization , for specifying what to do after object is about to be destroyed by garbage collector

Packages are containers of classes

Package is just a folder

### 1. What is Object-Oriented Programming (OOP)?

- Answer: OOP is a paradigm that organizes code into objects containing both data (attributes) and behavior (methods). It enables better code organization and modularity.

- Example: A "BankAccount" class with attributes `balance` and methods like `deposit()` and `withdraw()` is an OOP approach.

### 2. Define Class and Object.

- Answer: A class is a blueprint for creating objects. An object is an instance of a class, representing a specific entity with the attributes and behaviors defined by the class.

- Example: 'Car' can be a class, while 'myCar' with properties like color and model is an object.

### 3. What is Encapsulation?

- Answer: Encapsulation binds data and methods within a class and restricts access to some components, providing data protection.

- Example: A "User" class can have private attributes like `password` accessed only through secure methods.

#### 4. What is Inheritance?

- Answer: Inheritance allows a new class to inherit properties and methods from an existing class, promoting code reuse.
- Example: A `Bird` class with a `fly()` method could be a superclass for `Parrot` and `Sparrow`, which inherit `fly()`.

#### 5. Explain Polymorphism.

- Answer: Polymorphism allows methods to do different things based on the object it is acting upon, promoting flexibility.
- Example: A method `draw()` can work differently when called by objects of `Circle` and `Rectangle` classes.

#### 6. What is Abstraction?

- Answer: Abstraction hides complex implementation details and shows only the essential features.
- Example: A coffee machine user interacts only with the buttons without seeing the internal mechanism.

#### 7. What is a Constructor?

- Answer: A constructor initializes an object's properties when the object is created.
- Example: A "Book" class constructor might set `title` and `author` when creating a book object.

8. Explain Method Overloading and Method Overriding.

- Answer: Overloading allows multiple methods with the same name but different parameters. Overriding redefines a superclass method in a subclass.
- Example: A `print()` function could be overloaded for different data types; subclasses of `Animal` may override `speak()` differently.

9. What is the purpose of the `this` keyword?

- Answer: `this` refers to the current instance of the class, resolving conflicts between instance variables and method parameters.

10. What is a Static Method?

- Answer: A static method belongs to the class, not instances, and can be called without creating an object.

11. What are Access Modifiers?

- Answer: Access modifiers control the visibility of class members. Common modifiers include public, private, and protected.

12. Explain the concept of an Abstract Class.

- Answer: An abstract class cannot be instantiated and often contains abstract methods that subclasses must implement.
- Example: An `Animal` abstract class might require `move()` and `sound()` methods for its subclasses.

13. What is an Interface?

- Answer: An interface defines methods that implementing classes must provide, supporting abstraction and multiple inheritance.
- Example: A "Flyable" interface might require an `ascend()` and `descend()` method.

14. What is a Virtual Function?

- Answer: A virtual function allows derived classes to override a method from a base class, enabling runtime polymorphism.

15. What is the difference between Early Binding and Late Binding?

- Answer: Early binding resolves method calls at compile-time, while late binding (dynamic binding) resolves them at runtime.

16. Explain Association.

- Answer: Association is a relationship between two classes without ownership, meaning they can exist independently.
- Example: A "Teacher" and "Student" can have an association without one containing the other.

17. What is Aggregation?

- Answer: Aggregation is a form of association where one class has a relationship with another, but both can exist independently.
- Example: A "Library" and "Book" have an aggregation relationship, as books exist independently of the library.

18. Explain Composition.

- Answer: Composition is a strong form of association where one class cannot exist without the other.
- Example: A "House" and "Room" have a composition relationship; rooms are part of a house and cannot exist independently.

**19. What is a Singleton Class?**

- Answer: A singleton class restricts the instantiation of a class to one object. It is useful for managing shared resources.

**20. Explain the `super` keyword.**

- Answer: `super` refers to the superclass and is used to call superclass methods or constructors from a subclass.

**21. What is the difference between Interface and Abstract Class?**

- Answer: Abstract classes can have defined methods and support single inheritance, while interfaces require all methods to be abstract (until recent language updates) and allow multiple inheritance.

**22. What is Method Overriding?**

- Answer: Method overriding allows a subclass to provide a specific implementation of a method already defined in its superclass.

**23. What is a Final Class?**

- Answer: A final class cannot be subclassed, ensuring its behavior remains unchanged by any child class.

**24. Explain Object Cloning.**

- Answer: Object cloning creates a copy of an object, useful when you need a new object with the same state.

### 25. What is a Marker Interface?

- Answer: A marker interface has no methods but provides metadata to classes that implement it, like `Serializable`.

### 26. What is a Factory Design Pattern?

- Answer: The factory pattern provides an interface for creating objects without specifying their exact class, helping manage object creation logic.

### 27. What is the Observer Design Pattern?

- Answer: The observer pattern defines a one-to-many relationship, notifying dependent objects of state changes.
- Example: In a social media app, followers receive updates whenever the person they follow posts new content.

### 28. Explain Garbage Collection.

- Answer: Garbage collection automatically frees memory by deleting objects that are no longer referenced, preventing memory leaks.

### 29. What is Thread Safety in OOP?

- Answer: Thread safety ensures that a program behaves correctly when accessed by multiple threads, often requiring synchronization.

## 2. What are the main principles of OOPS?

- ✓ Abstraction – Exposing only the relevant details of an entity
- ✓ Encapsulation – Binding data and operations together in an entity
  - *Information hiding - Access control modifiers*
  - *Implementation hiding - through creation of interface for class*
- ✓ Inheritance – Reusability of code
- ✓ Polymorphism – One name, many forms ✓
  - *Static Polymorphism - Method overloading*
  - *Dynamic Polymorphism - Method overriding*

**CareerRide.com**

## 4. Constructor and Destructor

- ✓ **Constructor** – initializes the state of an object
  - Invoked at the time of object creation ✓
  - Instance method with usually the same name as class
- ✓ **Destructor** – Automatically called at the end of lifetime of an object
  - Freed up the acquired resources

## 6. What is Operator Overloading?

- Type of polymorphism
- Operator overloaded to give it user defined meaning

## 7. What is Method Overriding?

Redefining a super class method in a sub class

## 8. What is an abstract class?

- ✓ Process of hiding the implementation details while showing the functionality is Abstraction *Can have abstract & non-abstract methods*
- ✓ Class declared as Abstract – Abstract Class
- ✓ Needs to be extended and methods implemented
- ✓ Can't be instantiated

## 9. What is an interface?

- ✓ Specifies what a class must do
- ✓ Methods declared in interface are 'abstract' by default
- ✓ Way to achieve total abstraction

## 10. What is a virtual function?

- ✓ Member function declared within base-class and redefined by derived class
- ✓ Used to override a behavior for derived class at run-time

## 11. Early binding vs Late binding

Early binding	Late binding
Compile time binding ✓	Dynamic binding/ Run-time binding
Methods, functions and properties detected and checked during compile time	Methods, functions and properties detected and checked only at run-time
Performance and ease of development	

## 12. What is a copy constructor?

- ✓ Creates an object by initializing it with another object of same class.
- ✓ New object is exact copy of the existing one.

Q

What are the advantages and limitations of OOPS?

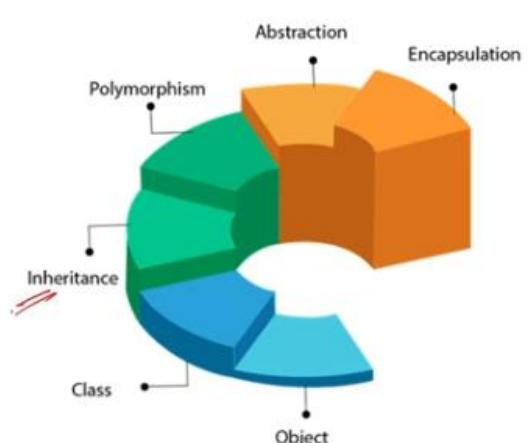


❖ Advantages of OOPS:

1. Reuse of code using inheritance.
2. Flexibility of code using polymorphism.
3. Secure application by using Encapsulation.
4. Easily scalable from small to large applications.
5. Easier troubleshooting of code because of modularity.

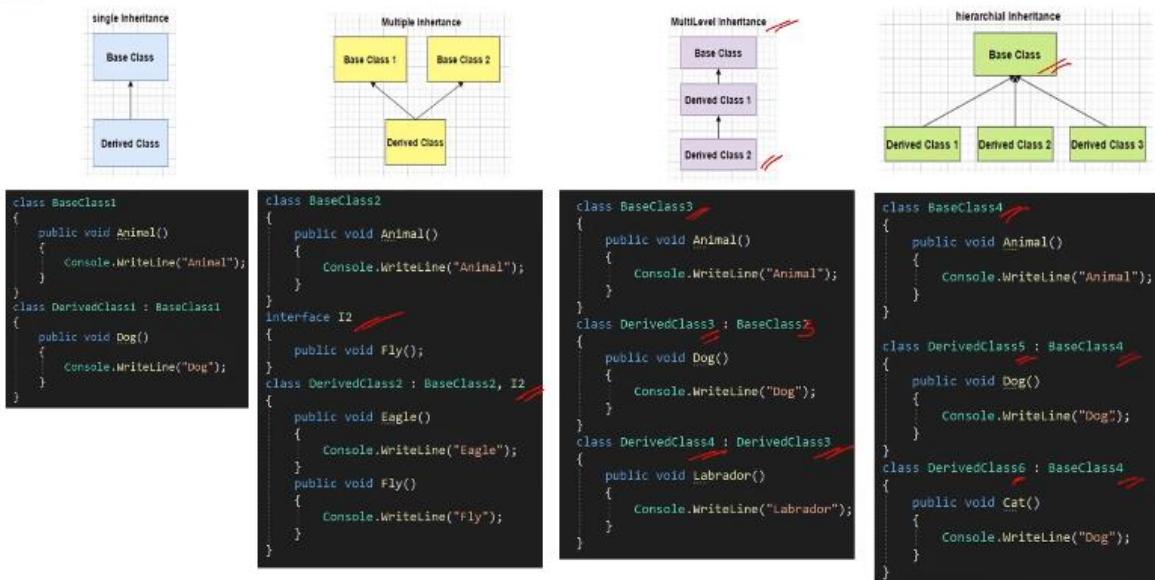
❖ Disadvantage of OOPS:

1. It is not suitable for small applications.



Q

What are the different types of Inheritance? V. IMP.



Q

What is Polymorphism and what are its types?



- ❖ Polymorphism is the ability of a variable, object, or function to take on **MULTIPLE FORMS**.

For example, in English “RUNNING” word can be used for “running a race” or “running a business”. In both cases the meaning is different.



```

public class Polymorphism
{
    public int Add(int a, int b)
    {
        return a + b;
    }

    public string Add(string str1, string str2)
    {
        return str1 + str2;
    }
}

```

Function with same name but have multiple forms

```

static void Main(string[] args)
{
    Polymorphism obj = new Polymorphism();

    int i = obj.Add(50, 60);
    string str = obj.Add("Interview", "Happy");

    Console.WriteLine(i + " - " + str);

    Console.ReadLine();
}

```

What is Method Overloading? In how many ways a method can be overloaded? **V. IMP.**



- Method overloading is a type of polymorphism in which we can create multiple methods of the **same name in the same class**, and all methods work in different ways.

```
public class MethodOverloading
{
    public int Add(int a, int b)
    {
        return a + b;
    }

    public int Add(int a, int b, int c)
    {
        return a + b + c;
    }

    public double Add(double a, double b, int c)
    {
        return a + b + c;
    }

    public double Add(double a, int c, double b)
    {
        return a + b + c;
    }
}
```

1. Number of parameters are different

2. Type of parameters are different

3. Order of parameters is different

Q

What is the difference between Overloading and Overriding? **V. IMP.**



#### Method Overloading

- Multiple methods of same name in **single class**.
- No need of **inheritance**, as it is in single class.
- All methods have **different signature**.
- It's a **compile time** polymorphism.
- No special keyword used.

```
public class MethodOverloading
{
    public int Add(int a, int b)
    {
        return a + b;
    }

    public int Add(int a, int b, int c)
    {
        return a + b + c;
    }

    public double Add(double a, double b, int c)
    {
        return a + b + c;
    }

    public double Add(double a, int c, double b)
    {
        return a + b + c;
    }
}
```

**Q** What is the difference between Overloading and Overriding? **V. IMP.**



❖ Method Overriding

1. Multiple methods of same name in **different class**.
2. **Inheritance is used**, as it is in different class.
3. All methods have **same signature**.
4. It's a **run time** polymorphism.
5. **Virtual & override** keywords.

```
public class BaseClass
{
    public virtual void Greetings()
    {
        Console.WriteLine("BaseClass Hello!");
    }
}

public class DerivedClass : BaseClass
{
    public override void Greetings()
    {
        Console.WriteLine("DerivedClass Hello!");
    }
}

static void Main(string[] args)
{
    DerivedClass objDerived = new DerivedClass();
    objDerived.Greetings();
    Console.ReadLine();
}

//Output: DerivedClass Hello
```

**Q** What is the difference between Method Overriding and Method Hiding?



```
public class BaseClass
{
    public virtual void Greetings()
    {
        Console.WriteLine("BaseClass Hello!");
    }
}

public class DerivedClass : BaseClass
{
    public override void Greetings()
    {
        Console.WriteLine("DerivedClass Hello!");
    }
}

static void Main(string[] args)
{
    BaseClass objDerived = new DerivedClass();
    objDerived.Greetings();
    Console.ReadLine();
}

//Output: DerivedClass Hello
```

```
public class BaseClass
{
    public void Greetings()
    {
        Console.WriteLine("BaseClass Hello!");
    }
}

public class DerivedClass : BaseClass
{
    public new void Greetings()
    {
        Console.WriteLine("DerivedClass Hello!");
    }
}

static void Main(string[] args)
{
    BaseClass objDerived = new DerivedClass();
    objDerived.Greetings();
    Console.ReadLine();
}

//Output: BaseClass Hello
```

| **What is the difference between Method Overriding and Method Hiding?**

- ❖ In Method Hiding, you can completely hide the implementation of the methods of a base class from the derived class using the **new keyword**.

```
public class BaseClass
```

```
public cla
```

Q

What is the difference between an Abstract class and an Interface (atleast 4)?

V. IMP.



1. Abstract class contains both **DECLARATION** & **DEFINITION** of methods.

```
public abstract class Employee
{
    public abstract void Project();
    public void Role()
    {
        Console.WriteLine("Engineer");
    }
}
```

Method Declared  
Method Defined

1. Interface should contain **DECLARATION** of methods.

❖ With C# 8.0, you can now have default implementations/definition of methods in an interface. But that is recommended in special case\*.

```
interface IEmployee
{
    public void Project1();
    public void Manager1();
}
```

Only method Declaration is allowed

2. Abstract class keyword: **ABSTRACT**

2. Interface keyword: **INTERFACE**

What is the difference between an Abstract class and an Interface (atleast 4)?

V. IMP.

3. Abstract class does not support **multiple inheritance**

```
public abstract class Employee
{
    public abstract void Project();
    public void Role()
    {
        Console.WriteLine("Engineer");
    }
}

public abstract class Employee1
{
    public abstract void Project1();
    public void Role1()
    {
        Console.WriteLine("Engineer1");
    }
}

public class PermanentEmployee: Employee, Employee1
{ }
```

3. Interface supports **multiple inheritance**.

```
interface IEmployee1
{
    public void Project1();
}

interface IEmployee2
{
    public void Project2();
}

public class NewEmployee : IEmployee1, IEmployee2
{
    public void Project1()
    {
        Console.WriteLine("Print 1");
    }

    public void Project2()
    {
        Console.WriteLine("Print 2");
    }
}
```

 What is the difference between an Abstract class and an Interface (atleast 4)? **V. IMP.**

4. Abstract class can have **constructors**.

```
public abstract class Employee1
{
    public Employee1()
    {
    }

    public abstract void Project1();

    public void Role1()
    {
        Console.WriteLine("Engineer1");
    }
}
```

4. Interface do not have constructors.

```
interface IEmployee1
{
    public IEmployee1()
    {
    }

    public void Project1();
}
```

## 1. Process States & PCB

Suggested: Lec-123: Top 10 SQL Interview

- A process control block (PCB) is a data structure used by computer operating systems to store all the information about a process.
- Process states: New, Ready, Running, Wait/Block, Suspend Ready, Suspend Block/Wait, Terminated
- A thread has three states: Running, Ready, and Blocked.

## 2. Mutex vs Binary Semaphore??

*Locks*

A Mutex is different than a semaphore as it is a locking mechanism while a semaphore is a signaling mechanism.

A binary semaphore can be used as a Mutex but a Mutex can never be used as a semaphore.

## 3. Process vs Thread

- Processes are basically the programs that are dispatched from the ready state and are scheduled in the CPU for execution.

Thread is the segment of a process which means a process can have multiple threads and these multiple threads are contained within a process.

## 4. Monolithic vs Microkernel

- The microkernel runs user and kernel services in different address spaces. On the other hand, the monolithic kernel runs both kernel and user services in the same address space.
- In microkernels, only essential processes like IPC, memory management, and scheduling take place in kernel space.

## 5. What is System call??

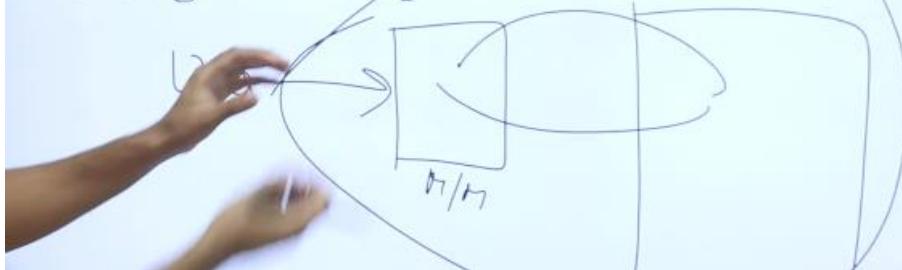
- A system call is a way for a user program to interface with the operating system. The program requests several services, and the OS responds by invoking a series of system calls to satisfy the request. Like: Fork(), Read(), Write(), Open(), Sleep() etc.

## 6. Demand Paging

- It says keep all pages of the frames in the secondary memory until they are required. It means do not load any page in the main memory until it is required.

## 7. What is Virtual Memory??

- Virtual Memory is a storage allocation scheme in which secondary memory can be addressed as though it were part of the main memory.

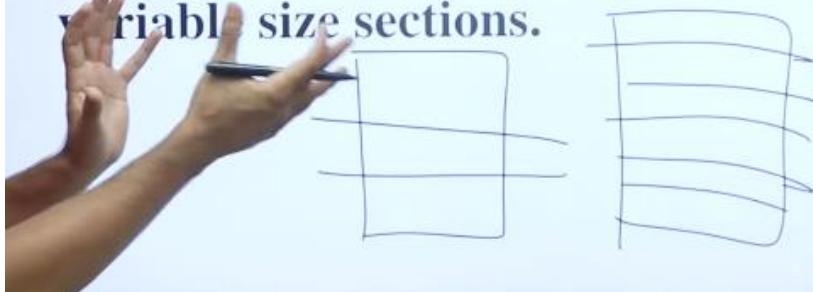


## 8. Chmod command

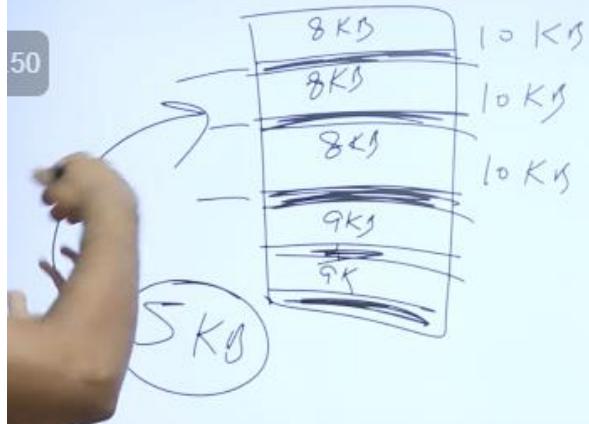
- chmod is the command and system call used to change the access permissions and the special mode flags of file system objects.

## 9. Paging vs Segmentation

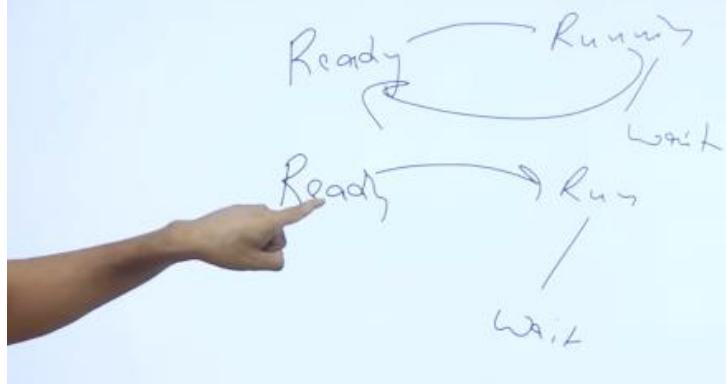
- In paging, the program is divided into fixed or mounted size pages.
- In segmentation, the program is divided into variable size sections.



## 10. Internal vs External fragmentation



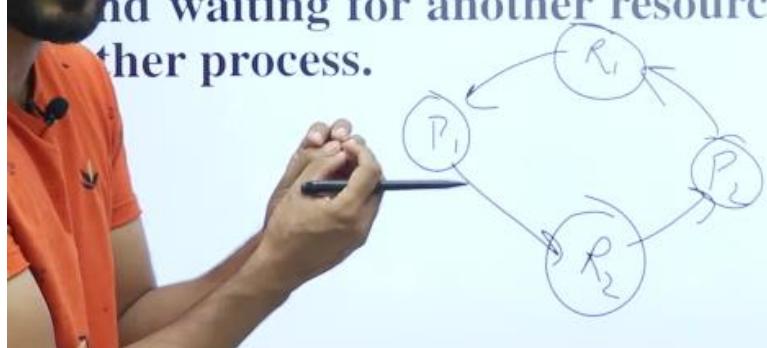
## 1. Preemptive vs Non Preemptive Scheduling



## 2. What is Deadlock?

1.50

Deadlock is a situation where a set of processes are locked because each process is holding a resource and waiting for another resource acquired by some other process.



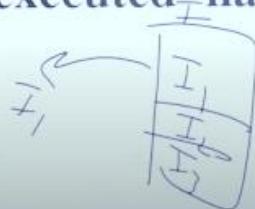
## 13. Multiprogramming vs Multitasking

- In multiprogramming, multiple programs execute at a same time on a single device.
- In Multitasking, a single resource is used to process multiple tasks.



## 14. Spatial Locality vs Temporal Locality

- Spatial Locality means that all those instructions which are stored nearby to the recently executed instruction have high chances of execution.
- Temporal Locality means that a instruction which is recently executed have high chances of execution again.

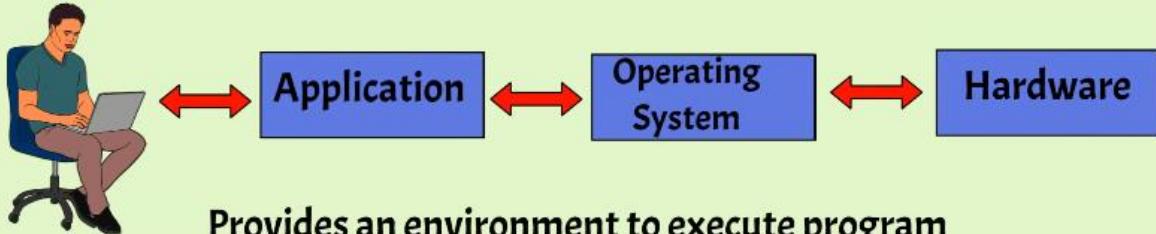


## 15. Virus, Worm, Trojan, Malware, Spyware

- "Malware" is short for malicious software and used as a single term to refer to virus, spy ware, worm etc. Malware is designed to cause damage to a stand-alone computer or a networked pc.
- Virus is a program written to enter to your computer and damage/alter your files/data. A virus might corrupt or delete data on your computer. Examples: W32.Sfc!mod, ABAP.Rivpas.A etc.
- Spyware is a type of program that is installed with or without your permission on your personal computers to collect information about users, their computer or browsing habits tracks.
- A Trojan horse is not a virus. It is a destructive program that looks as a genuine application. Unlike viruses, Trojan horses do not replicate themselves but they can be just as destructive.

Worms are malicious programs that make copies of themselves again and again on the local drive, network shares, etc. The only purpose of the worm is to reproduce itself again and again. It doesn't harm any data/file on the computer.

### 1. What is an **operating system**? What does it do?



**Provides an environment to execute program**

**Without OS, it would be just one dumb hardware machine**

**Memory management, File Management,  
I/O Management, Processor Management,  
Device Management, Security, Networking etc**

## 2. What are the most popular operating systems at present?

		iOS is for iPhone Owned by Apple 16% global share
<b>Operating System</b>		Microsoft

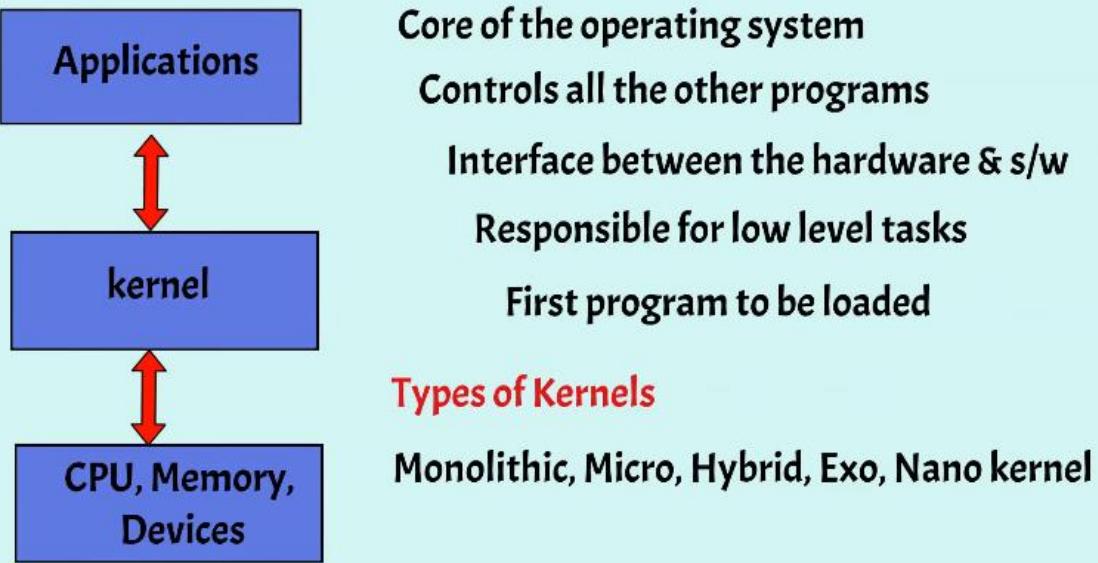
**World's most used mobile OS**  
**41% of the global market share**  
**Developed by Google**  
**Free & open-source software**

**2nd most used OS**  
**32% of the global share**  
**Owned by Microsoft**

## 3. What are the main functions of an operating system?

- It provides the interface**
- It organizes processing times of the various processes**
- It deals with memory management**
- It supervises scheduling of tasks and jobs**
- It also deals with various file activities**
- It also maintains security**

#### 4. What is a kernel? What are the different types of Kernels?



#### 5. What is the difference between Monolithic and Micro Kernel?

##### Monolithic Kernel

- User & the kernel services share the same memory space
- Execution of the process is faster
- Monolithic kernel is bigger in size
- Entire system fails when one service fails
- When a service is added, the entire OS needs to be modified
- The Linux kernel is an example of monolithic kernel

##### Micro Kernel

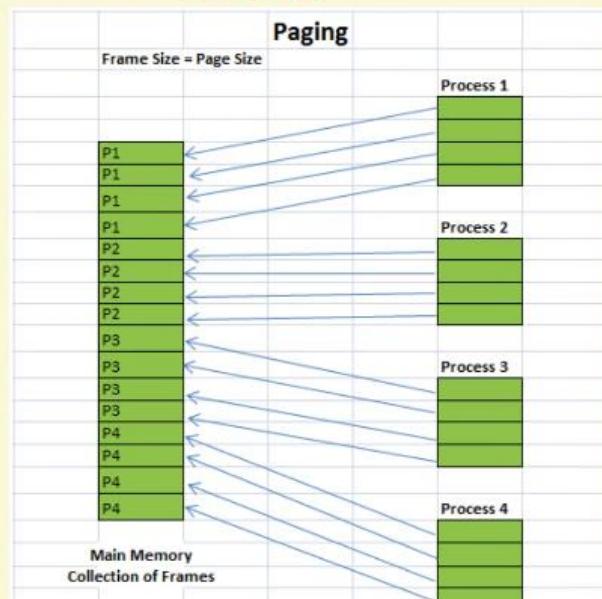
- User and kernel services are kept in separate address space
- Microkernel is smaller in size
- Operations are slower   It is easily extendible
- Minix, Symbian etc. are the examples microkernel

## 5. What are device drivers?



A device driver is a piece of software that allows the kernel of the OS to communicate with different hardware in the system.

## 7. What is paging?

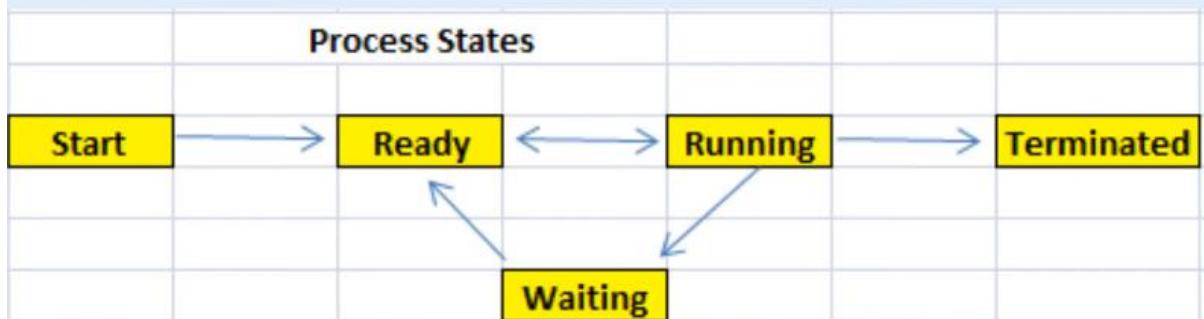


Paging is a memory management scheme

In paging, a process is divided into fixed-size contiguous block known as pages and is stored on the secondary storage

This process of transferring of pages between main memory and virtual memory is referred to as paging in Windows OS or swapping in Linux.

## 8. What is a **process** in operating system?



A process is a program in execution

When a process executes, it passes through many different stages like **Start, Ready, Running, Waiting** and **Terminated state**

## 9. What are the various **attributes** associated with a **process**?

Process Attributes		
Process Id		
Process State		
Priority		
Process Counter		
CPU registers		
List of open files		
List of open devices		

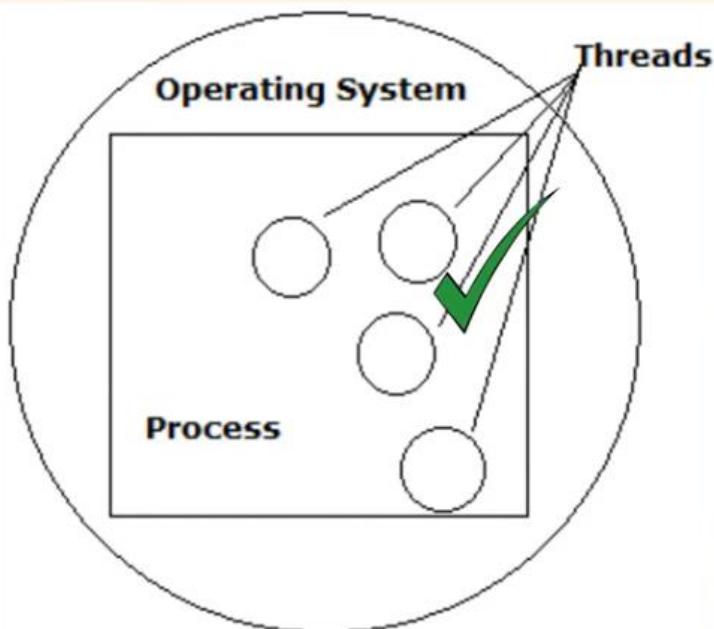
## 10. What is a **thread**?

A thread is the smallest unit of execution within a process

A thread is a lightweight process

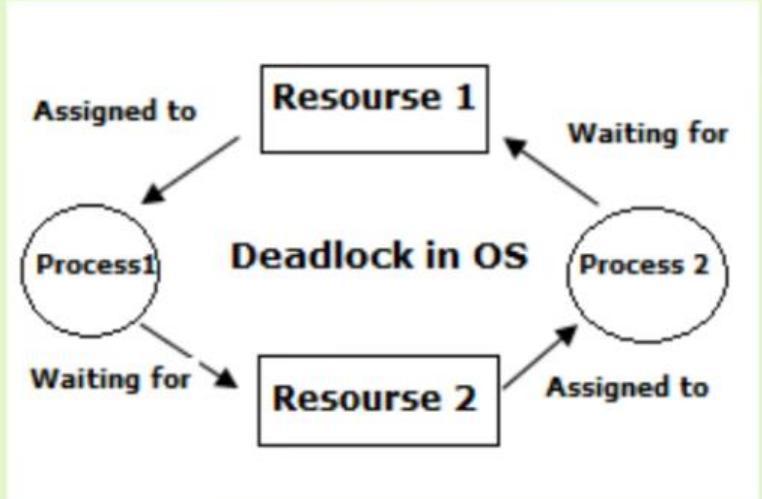
Threads enhance the throughput of the system

Multiple threads can run in parallel on different processors



## 11. What is a **deadlock**?

Deadlock is a situation where two processes are waiting for each other to complete in order to start their operation.



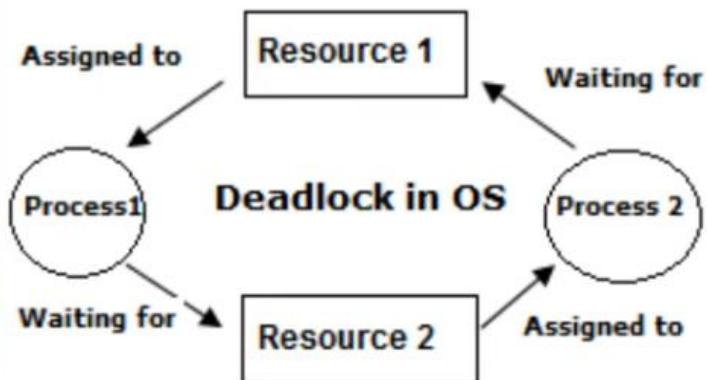
## 12. What are the necessary conditions that can lead to a deadlock situation in a system?

Mutual Exclusion

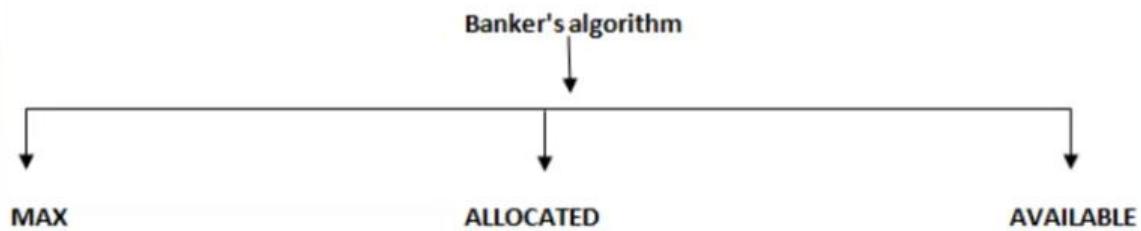
Hold and Wait

No Preemption

Circular wait



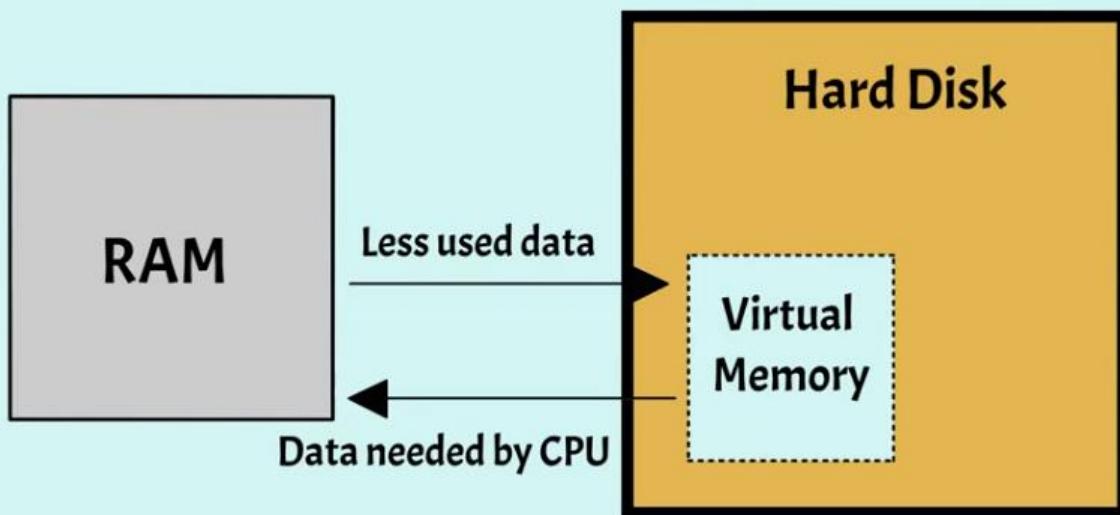
## 13. What is a Banker's algorithm?



Banker's algorithm is a deadlock avoidance algorithm

It is named as Banker's algorithm because the banks use the same technique to allocate money

## 14. What is a virtual memory?



## 15. What is demand paging?

Demand paging is a paging technique used in virtual memory system where the pages that are stored in the secondary storage are retrieved only when demanded by the CPU. It is also known as lazy swapper.

## 16. What is **thrashing**?

**Thrashing is a state where the CPU spends most of its time swapping pages between the main memory and virtual memory, rather than doing productive work such as executing the instructions.**

## 17. What is **page fault**?

**Page fault occurs when the page referenced by the CPU is not found in the main memory and has to be fetched from the secondary storage.**

## 18. What is the difference between **segmentation** and **paging**?

Paging	Segmentation
The size of a page is fixed. ✓	The size of a segment is not fixed. ✓
Procedures and data cannot be separated in paging	Procedures and data can be separated in segmentation.
It is faster	It is slower as compared to paging.
The size of the page is determined by the CPU and available memory.	The size of the segment is determined by the user.

## 19. What are the different states of a process?



## 20. What is a scheduling algorithm? Name the different types of scheduling algorithms.

A scheduling algorithm is an algorithm which intends to improve efficiency by reducing the waiting time to a minimum while allocating resources to various competing tasks.

## Operating Systems Important Questions and Answers

**Q: How does an OS act as an intermediary?**  
1.50

A: An OS functions as an intermediary between the user, applications, and hardware. It enables applications to request services like file access, memory usage, and network communication without dealing directly with the hardware. This abstraction allows the user and applications to operate at a high level, simplifying the use of complex hardware resources.

---

**Q: What are the primary functions of an OS?**  
1.50

A: The primary functions of an OS are managing hardware resources, handling I/O operations, memory management, file system organization, and user interaction. These functions make it easier to execute multiple applications simultaneously and protect system integrity. Through these tasks, the OS optimizes performance and ensures resources are used efficiently.

**Q: Explain the user view and system view of an OS.**

A: The user view of an OS focuses on providing a convenient and responsive environment to users, emphasizing features like a graphical interface, multitasking, and file management. The system view, on the other hand, prioritizes resource management, security, and stability. While the user view centers on ease of use, the system view addresses efficient hardware utilization.

---

**Q: What components are involved in a basic computer system?**

A: A basic computer system consists of the CPU (Central Processing Unit), memory, input/output (I/O) devices, and a system bus that connects these components. The CPU performs computations, memory stores data and instructions, and I/O devices enable interaction with the external environment. The bus facilitates data transfer between components, making up the core architecture for executing programs.

## Important Questions and Answers

---

**Q: How does the memory controller work in a computer system?**

A: The memory controller is a hardware component that manages access to memory, ensuring that memory access requests from the CPU and I/O devices are processed in an orderly fashion. It synchronizes data flow to prevent conflicts, particularly in multiprocessor systems. The OS works with the memory controller to allocate and control memory access for processes and devices.

---

**Q: What happens when a computer system starts?**

A: When a computer starts, it runs a bootstrap program stored in ROM or EEPROM, which initializes the OS and system hardware. This program loads the OS kernel into memory and hands control over to it. From there, the OS starts managing system resources and waits for user input or program requests, marking the beginning of standard operation.

---

**Q: What is an interrupt, and why is it important?**

A: An interrupt is a signal that causes the CPU to pause its current activities and execute a service routine, addressing an event such as input from an I/O device. Interrupts allow the OS to respond promptly to hardware events, like keyboard input, improving system responsiveness and efficiency by enabling the CPU to multitask.

**Q: What are the types of computer system architectures?**

A: Basic computer architectures include single-processor, multiprocessor, and clustered systems. A single-processor has one CPU, multiprocessors have multiple CPUs sharing tasks, and clustered systems connect independent computers working together. These architectures range from simple to highly parallel designs, allowing systems to scale in performance and resilience based on needs.

**Q:** Explain the difference between symmetric and asymmetric multiprocessing.

**A:** In symmetric multiprocessing (SMP), all processors have equal access to system resources and can run processes independently, improving scalability and efficiency. In asymmetric multiprocessing, a single master processor controls others, assigning tasks as needed. SMP offers balanced load sharing, while asymmetric multiprocessing provides simpler coordination.

**Q:** What is the main benefit of multiprocessor systems?

**A:** Multiprocessor systems increase computing power, reliability, and efficiency by distributing workloads among several CPUs. This enables parallel processing, where multiple processes can run simultaneously, reducing execution time and improving system throughput. Additionally, if one processor fails, others can continue, adding fault tolerance to the system.

**Q:** How do clustered systems work?

**A:** Clustered systems connect individual computers, or nodes, which work together to function as a single, cohesive system. These nodes share storage and are closely linked, often using high-speed connections. Clustered systems offer high availability and scalability by allowing tasks to be distributed across nodes and can recover from node failures, providing redundancy.

**Q:** What is a layered OS structure?

**A:** In a layered OS structure, the operating system is divided into layers, each layer performing specific tasks and providing services to the layer above. This modular approach simplifies OS design, as changes in one layer do not directly impact others, enhancing system maintainability and isolating functionalities for easier troubleshooting.

**Q:** Explain the microkernel approach.

**A:** A microkernel OS only includes essential functions, like process and memory management, in the kernel, with other functions running in user space. This separation minimizes the core kernel code, improving system stability and making it easier to modify or add new functions without affecting core operations, enhancing reliability.

**Q: What is the monolithic structure in an OS?**

A: A monolithic OS structure includes all essential services, like device drivers and file management, in a single, integrated kernel. While this can offer high performance, the lack of modularity means changes to one part can affect the entire kernel, making it harder to maintain and prone to system-wide errors if issues occur.

**Q: How does a modular OS structure differ from a monolithic one?**

A: In a modular OS, functions are implemented as separate modules that can be loaded and unloaded as needed. This structure provides flexibility, allowing for updates and functionality changes without rebooting or impacting the kernel as a whole. Modularity offers an advantage in handling complex tasks and managing diverse hardware.

**Q: What is dual-mode operation in an OS?**

A: Dual-mode operation separates user mode from kernel mode, allowing certain instructions to be executed only in kernel mode. This distinction prevents user applications from directly accessing hardware or system memory, enhancing security by restricting sensitive operations to the OS kernel, thus maintaining system stability and integrity.

---

**Q: How does an OS handle user and kernel mode transitions?**

A: The OS uses system calls to manage transitions from user to kernel mode, where protected operations are performed. When a user application needs kernel services, it issues a system call, switching control to the OS, which then executes the requested operation in kernel mode, protecting the system from potential security risks.

**Q: What is the purpose of virtual memory?**

A: Virtual memory expands a system's usable memory by using disk storage to simulate additional RAM. This allows larger programs to run than would fit in physical memory alone, enabling processes to operate smoothly without being constrained by the actual physical RAM available, providing flexibility and efficiency in memory management.

**Q: What is a process in an OS?**

**A:** A process is an active instance of a program, containing its own code, data, and state information like the program counter. Processes are fundamental units of execution, and the OS manages them to optimize resource use and allow multiple programs to run concurrently, ensuring system responsiveness.

**Q: Explain process scheduling.**

**A:** Process scheduling is the OS mechanism that decides which process runs at a given time, based on priority and resource availability. It is essential for balancing workloads, ensuring fair CPU allocation, and preventing processes from monopolizing system resources, making it a key part of multitasking.

**Q: What are the states of a process?**

**A:** Processes typically transition through states like new, ready, running, waiting, and terminated. These states represent the lifecycle of a process, with the OS managing transitions based on the process's needs and system availability, ensuring efficient execution and optimal CPU utilization.

**Q: What is a context switch?**

**A:** A context switch occurs when the OS saves the current state of a process and loads the state of another process, allowing multitasking. This ensures that the CPU can switch between processes without losing progress, enabling efficient resource sharing and minimizing idle CPU time.

## **DAEMON – DISK AND EXECUTION MONITOR**

**Zombie process – when child process is still running while parent process is terminated**

**Pipe - Ipc mechanism when 2 process are required to communicate one way**

**Ipc – inter process communication**

**Semaphore** - A semaphore is a synchronization mechanism used in computing to manage access to shared resources by multiple processes or threads. It is often employed in concurrent programming to prevent race conditions and ensure that critical sections of code are executed safely.

- **Binary Semaphore:** Takes values 0 or 1, used for mutual exclusion.
- **Counting Semaphore:** Can take multiple values, allowing a specified number of accesses.

**Operations:**

- **P (Wait):** Decreases the semaphore value; blocks if it's zero.
- **V (Signal):** Increases the semaphore value; wakes up blocked processes if any.

**Critical section – only one process at a time**

# DBMS

DBMS – SOFTWARE THAT MANAGE DATABASE  
 DATABASE – GROUP OF INTERRELATED RECORDS

## ADVANTAGE:

1. CONTROL DATA REDUNDANCY
2. EASILY MAINTENANCE
3. REDUCE TIME
4. BACKUP

Activati  
Go to Set

## ATTRIBUTES

1. SIMPLE ATTRIBUTES
2. COMPOSITE ATTRIBUTE
3. MULTI VALUED ATTRIBUTE
4. DERIVED ATTRIBUTES
5. KEY ATTRIBUTES

(1)

ROLL NO.	DOB	AGE	NAME	MOBILE
1	12/04/2001	21	SHIVAM	9999999999
2	06/12/1999	23	SOMU	8888888888
3	04/02/2000	22	GOLU	7777777777

## KEYS IN SQL

1. SUPER KEY
2. CANDIDATE KEY
3. PRIMARY KEY
4. FOREIGN KEY
5. ALTERNATE KEY

ROLL NO	DOB	AGE	NAME	MOBILE	MOBILE
1	12/04/2001	21	SHIVAM	9999999999	9999999999
2	06/12/1999	23	SOMU	8888888888	8888888888
3	04/02/2000	22	GOLU	7777777777	7777777777

## COMMAND

1. DDL      data definition language structure DRCAT
2. DML      data manipulation language DINU
3. DCL      data control language grant revoke
4. TCL      transaction control language commit ROLL BACK SAVEPOINT
5. DQL      DATA QUERY LANGUAGE SELECT

## CONSTRAINTS IN SQL

1. NOT NULL
2. UNIQUE
3. PRIMARY KEYS
4. FOREIGN KEYS
5. CHECK

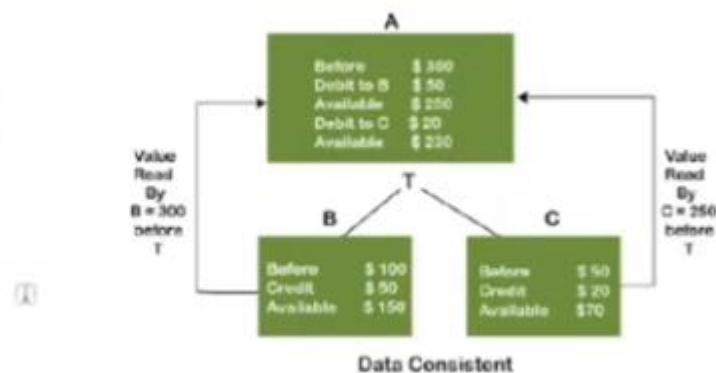
## CREATE COPY OF TABLE

DATA WITH SCHEMA -> `create table new select * from old;`

ONLY SCHEMA -> `create table new like old;`

## ACID PROPERTIES

- ATOMICITY
- CONSISTENCY
- ISOLATION
- DURABILITY



## DATA INTEGRITY

Data integrity refers to the overall accuracy, completeness, and reliability of data.

Employee_id	Name	Salary	Age
1	Andrew	486522	25
2	Angel	978978	30
3	Anamika	697abc	35

This value is out of domain(not INTEGER)so it is not acceptable.

## DOMAIN INTEGRITY

## DATA INTEGRITY

1. DOMAIN INTEGRITY - restricting the format, type, and volume of data recorded in a database
2. ENTITY INTEGRITY - The purpose is to ensure that data is not recorded multiple times

Employee_id	Name	Salary	Age
1	Andrew	486522	25
2	Angel	978978	30
3	Anamika	697abc	35

data integrity in a database

This value is out of domain(not INTEGER)so it is not acceptable.

## DOMAIN INTEGRITY

## DATA INTEGRITY

3. **REFERENCIAL INTEGRITY** - remove duplicate data records
4. User defined INTEGRITY-fulfill their specific requirements

Employee_id	Name	Salary	Age
1	Andrew	486522	25
2	Angel	978978	30
3	Anamika	697abc	35

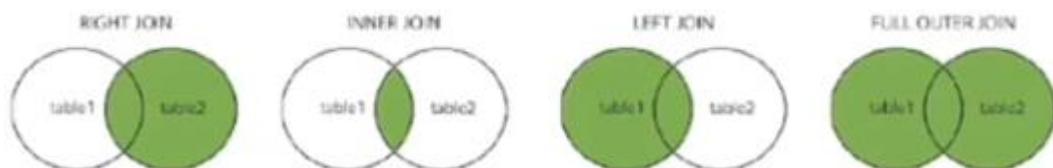
This value is out of domain(not INTEGER)so it is not acceptable.

## DOMAIN INTEGRITY

## Different Types of SQL JOINS

Here are the different types of the JOINs in SQL:

- **(INNER) JOIN:** Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN:** Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table



## NORMALISATION

TECHNIQUE TO REDUCE REDUNDANCY IN A TABLE.

**ANOMALIES:**

**INSERT**    **UPDATE**    **DELETE**

rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

## 1NF(1<sup>ST</sup> NORMAL FORM):

1NF – NO MULTIVALUED ATTRIBUTE

roll_no	name	subject
101	Akon	OS, CN
103	Ckon	Java
102	Bkon	C, C++

## 2NF(2nd NORMAL FORM):

2NF – 1NF & NO PARTIAL DEPENDENCY

DEPENDENCY: ALL COLUMN DATA CAN BE FETCHED FROM STUDENT\_ID.

Student\_id->name , student-id->reg\_no

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat

## 2NF(2nd NORMAL FORM):

2NF – 1NF & NO PARTIAL DEPENDENCY

the teacher's name only depends on subject, hence the subject\_id, and has nothing to do with student\_id.

This is Partial Dependency, where an attribute in a table depends on only a part of the primary key and not on the whole key.

score_id	student_id	subject_id	marks	teacher
1	10	1	70	Java Teacher
2	10	2	75	C++ Teacher
3	11	1	80	Java Teacher

## 3NF

**3NF- 2NF & NO TRANSITIVE DEPENDENCY**

score_id	student_id	subject_id	marks	exam_name	total_marks
1	001	CS1	85	ML	100
2	001	CS2	95	ADA	150
3	002	CS1	75	ML	100

**student\_id + subject\_id.** IS PRIMARY KEY

TOTAL MARKS IS NON PRIME ATTRIBUTE WHICH CAN BE DETERMINED BY ANOTHER NON PRIME ATTRIBUTE EXAM NAME.

This is **Transitive Dependency**. When a non-prime attribute depends on other non-prime attributes rather than depending upon the prime attributes or primary key.

## BCNF

**BCNF- 3NF & LHS OF FUNCTIONAL DEPENDENCY MUST BE PART OF SUPER KEY.**

student_id	subject	professor
101	Java	P.Java
101	C++	P.Cpp
102	Java	P.Java2
103	C#	P.Chash
104	Java	P.Java

there is one more dependency, **professor → subject**.

And while **subject** is a prime attribute, **professor** is a **non-prime attribute**, which is not allowed by BCNF.

## PATTERN MATCHING

LIKE clause is used for pattern matching task.

% → 0 or more character

\_ → 1 character

Q. Select name of student whose name starts with 'pa';

Select name from student where name like "pa%";

Q. Select name of student whose name starts with 'p' and contains 3 character in name;

Select name from student where name like "p\_\_";

## Character manipulation function

Upper()

Lower()

Initcap()

Length()

Concat()

**view**

A view is a virtual table based on the result set of an sql statement.

Create view abc as

Select column1,column2 from student where  
condition...

<b>DBMS Interview Questions</b>	
<b>Question 1</b>	<p>What are the differences between a DBMS and RDBMS?</p> <p><b>DBMS</b></p> <p>Provides an organized way of managing, retrieving and storing from a collection of logically related information.</p> <p><b>V/S</b></p> <p><b>RDBMS</b></p> <p>Provides the same as that of DBMS, but it provides with relational integrity.</p>

**DBMS Interview Questions**

**Question 2**

Explain the terms database and DBMS. Also, mention the different types of DBMS.

A software application which interacts with databases, applications, and users to capture and analyse the required data. The data stored in the database can be retrieved, deleted and modified based on the clients requirement.

HIERARCHIAL      NETWORK  
RELATIONAL      OBJECT-ORIENTED

[www.edureka.co](http://www.edureka.co)

**SUBSCRIBE**

## Database

A database is a place where data is stored and organized, so you can easily find and use it later.

### DBMS (Database Management System)

A DBMS is software that helps you manage and work with a database. It allows you to add, change, and retrieve data from the database.

**DBMS Interview Questions**

**Question 3**

What are the advantages of DBMS?

- 01 Data Independence
- 02 Sharing of Data
- 03 Integrity constraints
- 04 Redundancy control
- 05 Provide backup and recovery facility

## DBMS Interview Questions

### Question 4

Mention the different languages present in DBMS

## DDL

Consists of the commands that can be used to define the database schema.

## DML

Includes commands which deal with the manipulation of data present in database.

## DCL

Consists of commands which deal with the rights, permissions and other controls of the database system.

## TCL

Includes the commands which mainly deal with the transaction of database.

## DBMS Interview Questions

### Question 5

What do you understand by query optimization?

**Query optimization** is the phase which identifies a plan for evaluation query that has the least estimated cost.

This phase comes into the picture **when there are a lot of algorithms** and methods to execute the same task.

The advantages of query optimization are as follows:

01

**The output is provided faster**

02

**A larger number of queries can be executed in less time**

03

**Reduces time and space complexity**

## DBMS Interview Questions

### Question 6

Do we consider NULL values the same as that of blank space or zero?

**A NULL value is not at all same as that of zero or a blank space.**

NULL value **represents a value which is unavailable**, unknown, assigned or not applicable whereas a zero is a number and blank space is a character.

## DBMS Interview Questions

### Question 7

What do you understand by aggregation and atomicity?

## AGGREGATION

This property states that a database modification must either follow all the rules or nothing at all. So, if one part of the transaction fails, then the entire transaction fails

## ATOMICITY

This is a feature of the E-R model which allows a relationship set to participate in another relationship set.

## DBMS Interview Questions

### Question 8

What are the different levels of abstraction in the DBMS?

**Different levels of abstraction are as follows:**

Physical Level

Logical Level

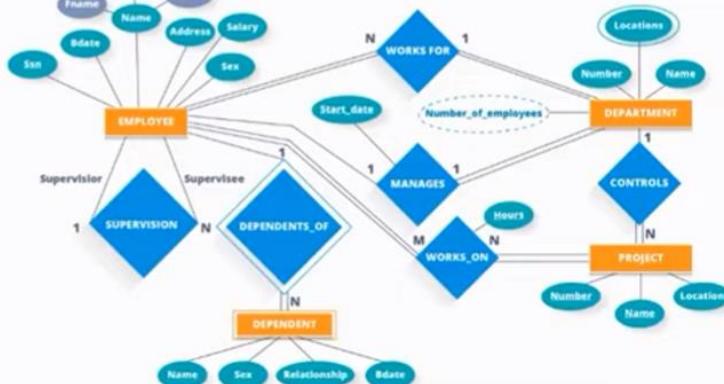
View Level

## DBMS Interview Questions

### Question 9

What is an entity-relationship model?

It is a diagrammatic approach to database design, where you represent real-world objects as entities and mention relationships between them. This approach helps the team of DBAs' to understand the schema easily.



## DBMS Interview Questions

### Question 10

What do you understand by the terms Entity, Entity Type, and Entity Set in DBMS?

#### Entity

An entity is a real-world object having attributes, which are nothing but characteristics of that particular object.

#### Entity Type

Entity type is nothing but a collection of entities, having the same attributes. Generally, an entity type refers to one or more related tables in a particular database.

#### Entity Set

An entity set is the collection of all the entities of a particular entity type in a database. For example, a set of employees, a set of companies, and a set of people can come under an entity set.

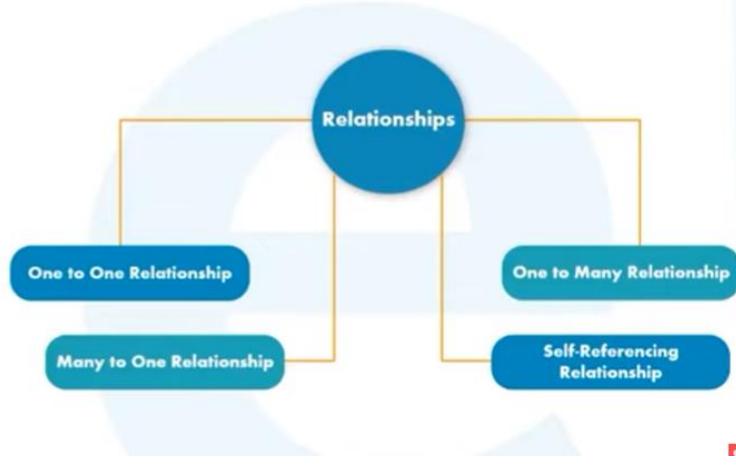
## DBMS Interview Questions

### Question 11

What are relationships and mention different types of relationships in the DBMS?

Relation or links are between entities that have something to do with each other.

Relationships are defined as the connection between the tables in a database.



## DBMS Interview Questions

### Question 12

What is concurrency control?

This is a process managing simultaneous operations in a database so that database integrity is not compromised.

The following are the two approaches involved in concurrency control:

**Optimistic approach**

**Pessimistic approach**

**DBMS**  
**Interview Questions****Question 13**

What are the ACID properties in DBMS?

**A**tomicity

Atomicity refers to those transactions which are completely successful or failed.

**C**onsistency

Consistency ensures that the data must meet all the validation rules.

**I**solation

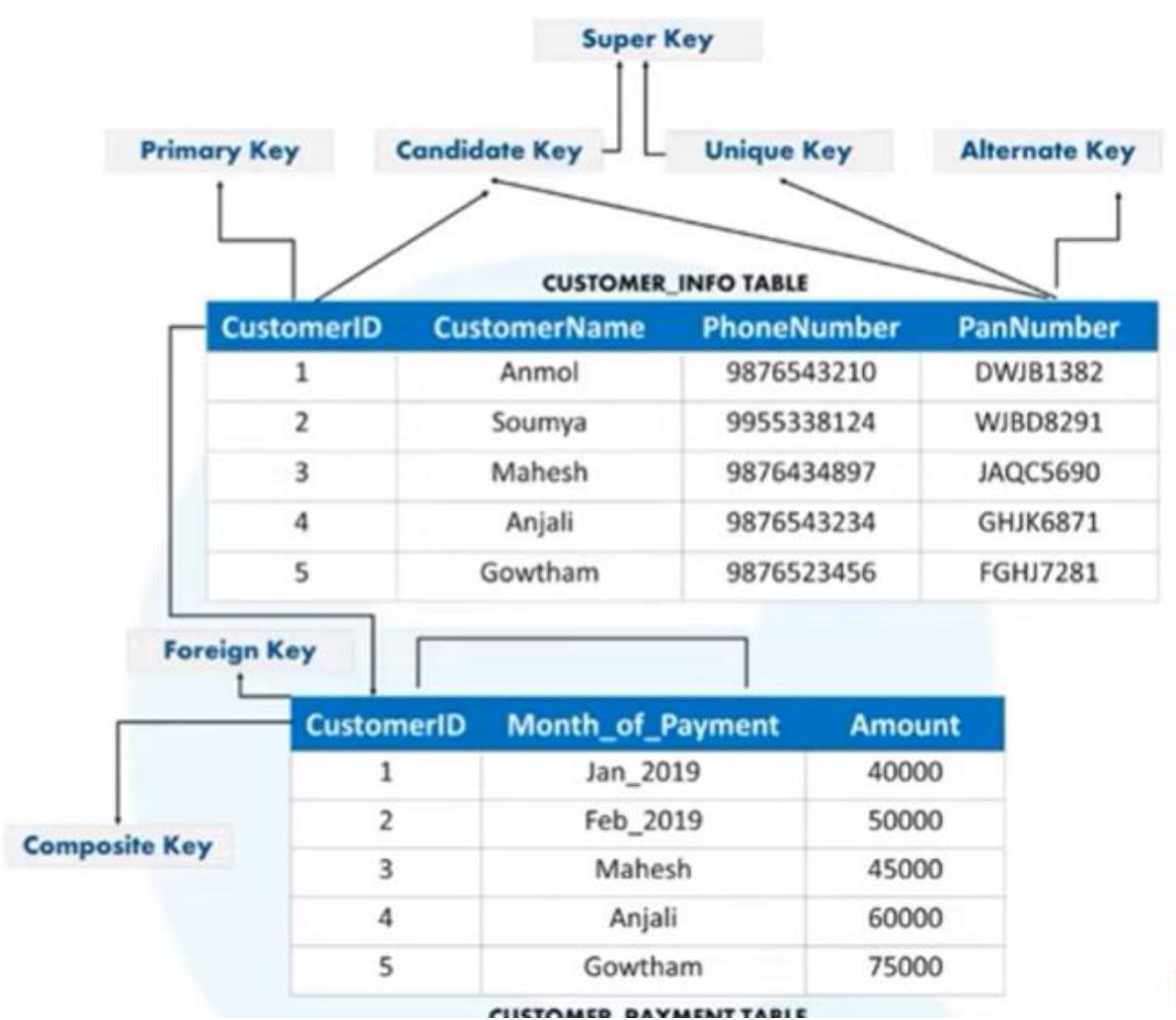
The main goal of isolation is concurrency control.

**D**urability

Durability means that if a transaction has been committed, it will occur whatever may be the scenario.

**BCNF**

If your database is in **3<sup>rd</sup> Normal Form**, there would be some scenarios where anomalies would be present, if you have more than candidate key. Then BCNF comes into role, where you divide your tables further so that there would be only one candidate key present.



50

## DBMS Interview Questions

### Question 16

What do you understand by correlated subqueries in DBMS?

A **correlated subquery** is also a sort of subquery reliant on another query.

Each subquery is executed a single time for every row of the outer query.

You can also understand correlated subqueries as those queries, which are used for row-by-row processing by the parent statement.

Here, the parent statement can be **SELECT, UPDATE or DELETE** statement.

## DBMS Interview Questions

### Question 17

Explain Database partitioning and its importance.

Data partitioning is the process of dividing a logical database into independent units for the betterment of availability, performance, and manageability.

O1 Enables you to access large parts of a specific partition

O2 Cheap and slower storage can be used to store data

O3 Improves query performance

1.50

## DBMS Interview Questions

### Question 18

What do you understand by functional dependency and transitive dependency in DBMS?

## FUNCTIONAL DEPENDENCY

A functional dependency is a constraint which is used in describing the relationship among different attributes in a relation.

### Example:

Consider a relation "A1" having attributes X and Y.

The functional dependency among these two attributes will be:

**X -> Y**

SU  
CR

## DBMS Interview Questions

### Question 18

What do you understand by functional dependency and transitive dependency in DBMS?

## DBMS Interview Questions

### Question 20

Mention the differences between Unique Key and Primary Key

## TRANSITIVE DEPENDENCY

A transitive dependency is a constraint which can only occur in a relation of three or more attributes.

### Example:

Consider a relation "A1" having attributes X, Y and Z. Now, X->Z is said to hold transitive dependency, only if the following functional dependencies holds true:

$$\begin{aligned} X &\rightarrow Y \\ Y &\text{ doesn't } \rightarrow X \\ Y &\rightarrow Z \end{aligned}$$

SI

www

## UNIQUE KEY

Unique Key can have a NULL value

Each table can have more than one unique key

## V/S

## PRI<sub>Y</sub>MA<sub>R</sub>Y KE<sub>Y</sub>

The primary key cannot have a NULL value

Each table can have only one primary key

- A candidate key is a potential primary key; there can be multiple candidate keys, but only one is chosen as the primary key.
- Unique keys are used to enforce uniqueness on non-primary key attributes.

## DBMS Interview Questions

### Question 21

What is a checkpoint in DBMS and when does it occur?

A **checkpoint** is a mechanism where **all the previous logs are removed from the system** and are permanently stored on the storage disk.

So, basically, checkpoints are **those points from where the transaction log record can be used to recover all the committed data up to the point of crash.**

## DBMS Interview Questions

### Question 22

Mention the differences between Trigger and Stored Procedures

### Triggers

A special kind of stored procedure that is not called directly by a user. In fact, a trigger is created and is programmed to fire when a specific event occurs.

### Stored Procedures

A group of SQL statements which can be reused again and again. These statements are created and stored in the database.

A trigger cannot be called or execute directly by a user. Only when the corresponding events are fired, triggers are created.

Can execute stored procedures by using the exec command, whenever we want.

You cannot schedule a trigger.

You can schedule a job to execute the stored procedure on a pre-defined time.

Triggers	Stored Procedures
Cannot directly call another trigger within a trigger.	Call a stored procedure from another stored procedure.
Parameters cannot be passed as input.	Parameters can be passed as input
Cannot return values.	Can return zero or n values.
Transactions are not allowed within a trigger.	You can use transactions within a stored procedure.

- **Trigger:** A trigger is a set of instructions that automatically executes (or "fires") in response to certain events on a specified table or view in a database. Common events that can trigger it include INSERT, UPDATE, or DELETE operations.
- **Stored Procedure:** A stored procedure is a precompiled collection of one or more SQL statements that can be executed as a single unit. It is called explicitly by a user or application and can take parameters for flexible functionality.

50

### DBMS Interview Questions

Question 23

What are the differences between Hash join, Merge join and Nested loops?

HASH JOIN	MERGE JOIN	NESTED LOOPS
The hash join is used when you have to join large tables.	Merge join is used when projections of the joined tables are sorted on the join columns.	The nested loop consists of an outer loop and an inner loop.

## DBMS Interview Questions

### Question 24

What do you understand by Proactive, Retroactive and Simultaneous Update?

### Proactive Update

These updates are applied to the database before it becomes effective in the real-world environment.

### Retroactive Update

These retroactive updates are applied to a database after it becomes effective in the real-world environment.

### Simultaneous Update

These updates are applied to the database at the same instance of time as it becomes effective in a real-world environment.

## DBMS Interview Questions

### Question 25

What are indexes? Mention the differences between the clustered and non-clustered index

**Indexes** are **data structures** responsible for improving the speed of data retrieval operations on a table.

This data structure **uses more storage space to maintain extra copies of data by using additional writes.**

Used for **searching algorithms**, where you wish to retrieve data in a quick manner.

## DBMS Interview Questions

### Question 25

indexes? Mention differences between the clustered and non-clustered index

Clustered Index	Non Clustered Index
A clustered index is faster	Non clustered index is relatively slower
Alters the way records are stored in a database as it sorts out rows by the column which is set to be clustered index	Does not alter the way it was stored but it creates a separate object within a table which points back to the original table rows after searching
One table can only have one clustered index	One table can only have many non clustered indexes

## DBMS Interview Questions

### Question 26

What do you understand by intension and extension?

## Intension

Intension or most commonly known as Database schema defines the description of the database. This is specified during the database design and mostly remains unchanged.

## Extension

Extension is the number of tuples available in the database at any instance of time. This value keeps changing as and when the tuples are created, updated and destroyed.

## DBMS Interview Questions

### Question 27

What do you understand by cursor? Mention the different types of cursor

A cursor is a database object which helps in manipulating data, row by row and represents a result set.

**Implicit cursor**

**Explicit cursor**

## DBMS Interview Questions

### Question 28

Explain the terms specialization and generalization

### Specialization

Specialization is a process of defining a set of subclasses of the entity type. Here, each subclass will contain all the attributes and relationships of the parent entity. Apart from this, the subclasses may contain additional attributes and relationships specific to itself.

### Generalization

Generalization is a process of finding relations, common attributes for a particular set of entities; and finally defining a common superclass for them.

## DBMS Interview Questions

### Question 29

What do you understand by Data Independence?

**When you say an application has data independence, it implies that the application is independent of the storage structure and data access strategies of data.**

**DBMS Interview Questions**

**Question 30**

What are the different integrity rules present in the DBMS?

The different integrity rules present in DBMS are as follows:

- 01 **Entity Integrity**
- 02 **Referential Integrity**

the value of primary key can never be null

Either the value of a foreign key is null or it should be the primary key of any other relation

**DBMS Interview Questions**

**Question 31**

What does Fill Factor concept mean with respect to indexes?

**Fill Factor** is used to mention the percentage of space left on every leaf-level page, which is packed with data.

Usually, the **default value** is 100.

## DBMS Interview Questions

### Question 32

What is Index hunting and how does it help in improving query performance?

The process of boosting a collection of indexes is known as Index hunting. This is done as indexes improve the query performance and the speed at which they are processed.

It helps in improving query performance in the following way:

The best queries are suggested using the query optimizer.

Index, query distribution and their performance are used as metrics to check the effect

Databases are tuned into a small collection of problem queries.

## DBMS Interview Questions

### Question 33

What are the differences between network and hierarchical database model?

#### Network Model

#### Hierarchical Model

Each parent node can have multiple children nodes and vice versa.

A top-down structure where each parent node can have many child nodes. But, a child node can have only a single parent node.

Supports one-to-one, one-to-many, and many-to-many relationships

Supports one-to-one and one-to-many relationships

## DBMS Interview Questions

### Question 34

Explain what is a deadlock and mention how it can be resolved?

**Deadlock** is a situation which occurs when two transactions wait on a resource which is locked or other transaction holds.

Deadlocks **can be prevented** by making all the transactions acquire all the locks at the same instance of time.

So, once deadlock occurs, the only way to cure is to **abort one of the transactions** and remove the partially completed work.

## DBMS Interview Questions

### Question 35

What are the differences between an exclusive lock and a shared lock?

## EXCLUSIVE LOCKS

An exclusive lock is a lock on a data item when a transaction is about to perform the write operation.

**V/S**

## SHARED LOCKS

A shared lock allows more than one transaction to read the data items.

## DBMS Interview Questions

### Question 1

What are the differences between DROP, TRUNCATE and DELETE commands?

DROP	TRUNCATE	DELETE
Used to delete a database, table or a view	Used to delete all rows from a table	Used to delete a row in the table
Data cannot be rolled back	Data cannot be rolled back	Data can be rolled back
A DDL command	A DDL command	A DML command.
Slower than TRUNCATE	Faster than DROP and DELETE	Slower than TRUNCATE
Deletes the full structure of the table	Preserves the structure of the table	Deletes the structure of the row from a table

## DBMS Interview Questions

### Question 3

What do you understand by sub-queries in SQL?

A subquery is **a query inside another query** where a query is defined to retrieve data or information back from the database.

In a subquery, the outer query is called as the **main query** whereas the **inner query** is called subquery.

Subqueries are always executed first and the result of the subquery is passed on to the main query.

It can be nested inside a **SELECT, UPDATE** or any other query.

A subquery can also use any comparison operators such as **>,< or =**.

## DBMS Interview Questions

### Question 4

Mention the differences between UNION and UNION ALL

## UNION

## UNION ALL

Combines the result of two or more SELECT statements consisting of distinct values

Combines the result set of two or more SELECT statements consisting of duplicate values

**Syntax:** UNION

**Syntax:** UNION ALL

Has low performance than UNION ALL, as duplicate rows need to be removed.

Has better performance than UNION, as duplicate rows need not have to be removed.

**DBMS Interview Questions**

**Question 5**

What do you understand by CLAUSE in SQL?

**CLAUSE in SQL** is used to **limit the result set** by mentioning a condition to the query. So, you can use a CLAUSE to filter rows from the entire set of records.

**Example:** WHERE, HAVING clause.

**DBMS Interview Questions**

**Question 6**

Mention the differences between HAVING and WHERE clause?

## HAVING CLAUSE

Can be used only with SELECT statement. It is usually used in a GROUP BY clause.

## WHERE CLAUSE

WHERE Clause is applied to each row before they are a part of the GROUP BY function in a query.

**DBMS Interview Questions**

**Question 7**

How can you perform pattern matching in SQL?

**LIKE operator is used for pattern matching.**

**% – It matches zero or more characters.**

**Example :**

```
SELECT * FROM students WHERE studentname LIKE
'a%'
```

**\_ – It matches exactly one character.**

**Example :**

```
SELECT * FROM student WHERE studentname LIKE
'abc_'
```

**DBMS Interview Questions**

**Question 8**

Mention few case manipulation functions in SQL

**LOWER**

This function returns the string in lowercase. It takes a string as an argument and returns it by converting it into lower case.

**Syntax:** LOWER('string')

**UPPER**

This function returns the string in uppercase. It takes a string as an argument and returns it by converting it into uppercase.

**Syntax:** UPPER('string')

**INITCAP**

This function returns the string with the first letter in uppercase and rest of the letters in lowercase.

**Syntax:** INITCAP('string')

**DBMS Interview Questions**

**Question 10**

What do you understand by the view and mention the steps to create, update and drop a view?

A view is a virtual table which consists of a subset of data contained in a table. Since views are not present, it takes less space to store. View can have data of one or more tables combined and it depends on the relationship.

**DBMS Interview Questions**

**Question 3**

Write a query to remove duplicate rows from a table?

50 **DBMS Interview Questions**

**Question 4**

Mention a query to add email validation to your database

To remove duplicate rows from a table, you have to initially select the duplicate row from the table without using the **DISTINCT keyword**.

So, to select the duplicate rows from the table, you can write a query as follows:

```
SELECT CustomerNumber FROM Customers WHERE
ROWID (SELECT MAX (rowid) FROM Customers C
WHERE CustomerNumber = C.CustomerNumber);
```

Now, to delete the duplicate records from the Customers table, mention the following query:

```
DELETE FROM Customers WHERE ROWID(SELECT MAX
(rowid) FROM Customers C WHERE CustomerNumber =
C.CustomerNumber);
```

Well, there are multiple ways to add email validation to your database, but one out of the lot is as follows:

```
SELECT Email FROM Customers WHERE NOT
REGEXP_LIKE(Email, '[A-Z0-9._%+-]+@[A-Z0-9.-
].[A-Z]{2,4}', 'i');
```

## DBMS Interview Questions

### Question 5

Write a query to retrieve last day of next month in Oracle.

To write a query to retrieve the last day of the next month in Oracle, you can write a query as follows:

```
SELECT LAST_DAY (ADD_MONTHS (SYSDATE,1)) from dual;
```

01

## DBMS Interview Questions



What is data?

- Data is stored facts.
- It can be numbers, words, measurements, observations or even just descriptions of things.



02

## DBMS Interview Questions



What is information?

- Information is that which informs, i.e. that from which data can be derived.
- Information is conveyed either as the content of a message or through direct or indirect observation of something.

03

## DBMS Interview Questions



What is Database?

- In simple words, Database is a collection of data in some organized way to facilitate its user's to easily access, manage and upload the data.



04

## DBMS Interview Questions



What is DBMS and RDBMS?

- DBMS (Database Management System) is software for storing and retrieving users' data while also considering appropriate security measures.
- RDBMS is a short form of Relational Database Management system. It is a software system that stores only data that needs to be stored in related tables.

09

## DBMS Interview Questions



What are indexes?

- A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data.

12

## DBMS Interview Questions



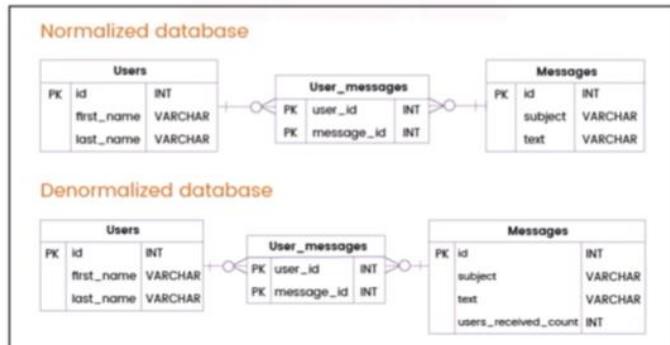
What is the purpose of normalization in DBMS?

- It is used to remove duplicate data and database anomalies from the relational table.
- Normalization helps to reduce redundancy and complexity by examining new data types used in the table.
- It is helpful to divide the large database table into smaller tables and link them using relationships.



## What is Denormalization?

- Denormalization is a database optimization technique in which we add redundant data to one or more tables.



## What do you mean by a database schema and a database state?

- The collection of information stored in a database at a particular moment in time is called database state while the overall design of the database is called the database schema.



## What is a view in SQL? How to create a view?

- A view is a virtual table based on the result-set of an SQL statement. We can create it using create view syntax.

### Syntax -

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```



## What are the uses of view?

- Views can join and simplify multiple tables into a single virtual table.
- Views can act as aggregated tables, where the database engine aggregates data (sum, average, etc.) and presents the calculated results as part of the data.
- Views can hide the complexity of data.



### What is Identity?

- Identity (or AutoNumber) is a column that automatically generates numeric values.
- A start and increment value can be set, but most DBA leave these at 1.

2.00

**18**

## DBMS Interview Questions



### What is QBE?

- Query-by-example represents a visual/graphical approach for accessing information in a database through the use of query templates called skeleton tables.
- It is used by entering example values directly into a query template to represent what is to be achieved.



### Explain different types of keys in a database.

- **Candidate Key:** The candidate key represents a set of properties that can uniquely identify a table.
- **Super Key:** The super key defines a set of attributes that can uniquely identify a tuple.
- **Unique Key:** The unique key is very similar to the primary key except that primary keys don't allow NULL values in the column but unique keys allow them.
- **Alternate Key:** All the candidate keys which are not chosen as primary keys are considered as alternate Keys.
- **Composite Key:** A composite key refers to a combination of two or more columns that can uniquely identify each tuple in a table.



### What is the difference between Trigger and Stored Procedure?

- A Trigger is a code associated with insert, update or delete operations.
- A stored procedure is like a function that contains a set of operations compiled together.
- Unlike Stored Procedures, Triggers cannot be called directly. They can only be associated with queries.



### What is a Live Lock?

- Live Lock situation can be defined as when two or more processes continually repeat the same interaction in response to changes in the other processes without doing any useful work.



### Why are cursors necessary in embedded SQL?

- A cursor is an object used to store the output of a query for row-by-row processing by the application programs.
- SQL statements operate on a set of data and return a set of data.



### What is Correlated Subquery in DBMS?

- A Subquery is also known as a nested query i.e. a query written inside some query.
- When a Subquery is executed for each of the rows of the outer query then it is termed as a Correlated Subq



### Explain Entity, Entity Type, and Entity Set in DBMS

- The entity is an object, place, or thing which has its independent existence in the real world and about which data can be stored in a database.
- Entity Type is a collection of entities that have the same attributes.
- Entity Set is a collection of entities of the same type.



What is BCNF in the DBMS?

- Any table is said to have in the BCNF if it satisfies the following 2 conditions:
- A table is in the 3NF.
- For each of the functional dependencies  $X \rightarrow Y$  that exists, X is the super key of a table.



What do you mean by Entity type extension?

- Compilation of similar entity types into one particular type which is grouped together as an entity set is known as entity type extension.

44

## DBMS Interview Questions



What is conceptual design in dbms?

- Conceptual design is the first stage in the database design process.
- The goal at this stage is to design a database that is independent of database software and physical details.



What are temporary tables? When are they useful?

- Temporary tables exist solely for a particular session, or whose data persists for the duration of the transaction.
- The temporary tables are generally used to support specialized rollups or specific application processing requirements.



What is the main goal of RAID technology?

- RAID technology was developed to address the fault-tolerance and performance limitations of conventional disk storage.
- It can offer fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives.

A redundant array of inexpensive disks

01

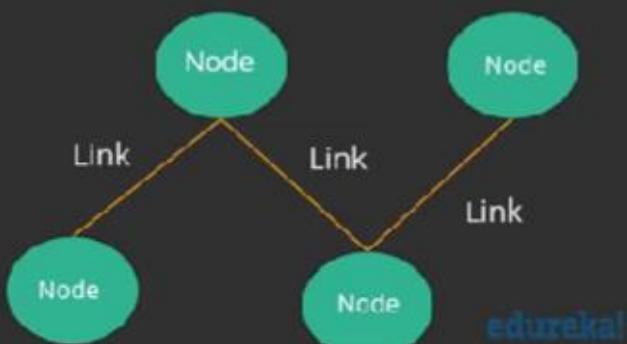
Differentiate between a router, a hub, and a switch.



HUB	SWITCH	ROUTER
Connects two or more Ethernet devices	Connects two or more LAN devices	Can connect devices or a LAN and WAN
Does not perform filtering	Filters packets before forwarding them	Highly configured to filter and send packets
Least intelligent, least expensive and least complex	Similar to a hub, but more effective	Extremely smart and complex

## What is a link?

A **link** is a connection between two or more computers or devices. It can be a physical connection or a wireless one. Physical links include cables, hubs, switches, etc and wireless links wireless access points, routers, etc.

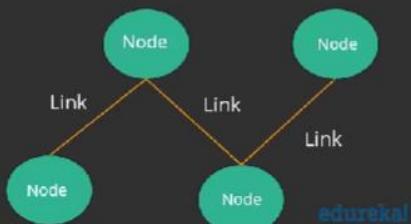


03

## What is a Node?



The point of intersection in a network is called a **Node**. Nodes can send or receive data/ information within a network.

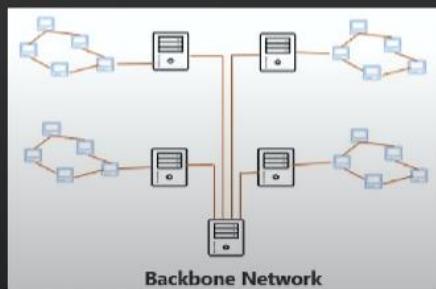


04

## What does a backbone network mean?

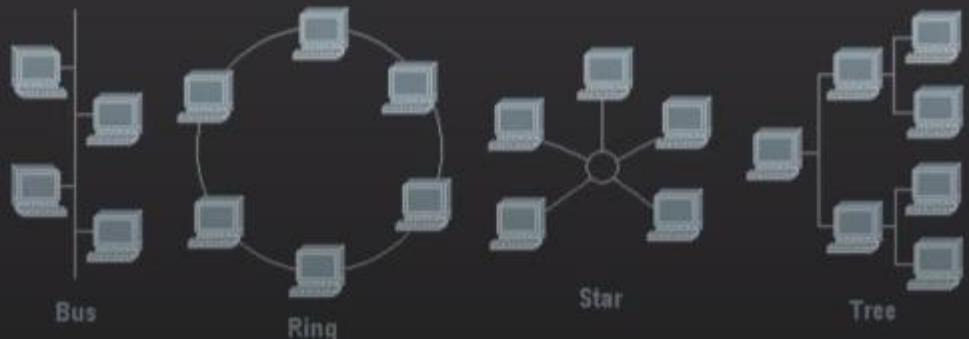


A **Backbone Network** is a Network that interconnects various parts of the network to which it belongs.



## What is Network Topology?

Physical layout of the computer network is called as Network Topology.



## Explain what is LAN?

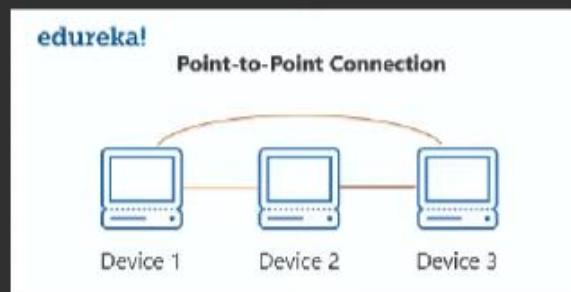
LAN is a wireless or wired network between devices located within a small physical location.

## What are Routers?

A router is a device that transfers the data packets within a network. A data packet can be anything such as an email, a web page, etc. Routers are located at the place where two or more networks meet.

## What is a Point-to-Point network?

A Point-to-Point network refers to a physical connection between two nodes. It can be between any device of a network such as a computer, printer, etc.



## What is OSI Model?

OSI stands for Open Systems Interconnection. It is a conceptual model that standardizes communication functions of telecommunication. It has 7 layers which are:

- Physical Layer
- Data Link Layer
- Network Layer
- Transport Layer
- Session Layer
- Presentation Layer
- Application Layer

Give a brief description about each layer in the OSI Model?

Layer	Protocol	Description
Physical Layer	Symbol	Transfers raw bits
Data link Layer	Frame	Transmission of data frames between nodes connected by physical layer
Network Layer	Packet	Structures and manages a network with multiple nodes
Transport Layer	Segment, Datagram	Transmission of data packets between different points of a network
Session Layer	Data	Manages the communication sessions
Presentation Layer	Data	Transmission of data between the service device and the application
Application Layer	Data	Specifies the shared communication protocols and the interface methods

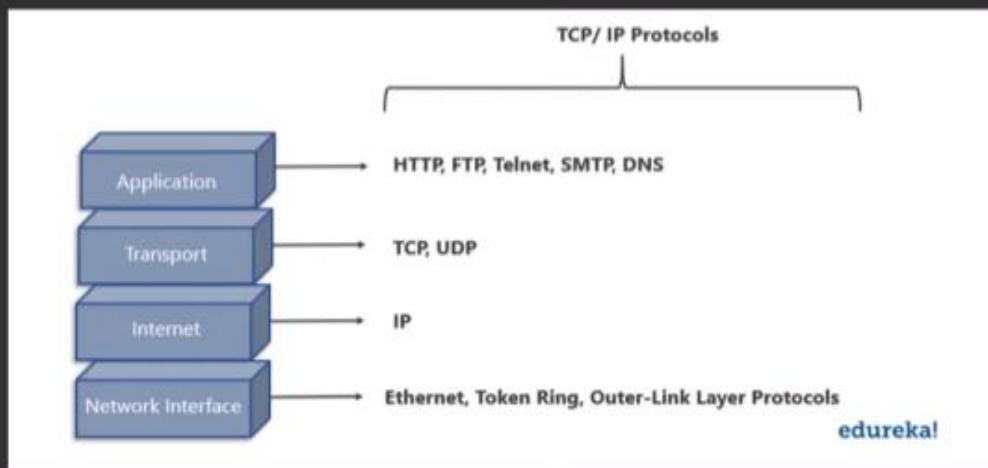
## What do you mean by anonymous FTP?

An anonymous FTP is a way of allowing a user to access data that is public. The user does not need to identify himself to the server and has to log in as anonymous.

## What do you mean by a Subnet Mask?

A Subnet Mask is the number describing the range of IP addresses that can be used within a network. They are used to assign subnetworks or subnets. These subnetworks are various LAN's connected to the internet.

## Give a brief description of the TCP/ IP Model.



## What is the difference between the OSI Model and TCP/ IP Model?

OSI Model	TCP/ IP Model
Has four layers	Has seven layers
More reliable	Less reliable
No strict boundaries	Has strict boundaries
Horizontal Approach	Vertical Approach

## What is a UTP cable?

- UTP cable is a 100 ohms cable made up of copper. It consists of 2-1800 unshielded twisted pairs that are surrounded by a non-metallic case. These twists provide immunity to electrical noise and EMI.

## Explain what is HTTP and which port does it use?

HTTP allows communication over the Internet. It defines how messages are to be transmitted and formatted over www.

Features of HTTP Protocol:

- It is connection-less
- Does not depend on the type of connecting media
- Stateless

## What is NAT?

NAT stands for Network Address Translation. It deals with remapping one IP Address space with another by changing the IP headers of the packets that are being transmitted across a traffic routing device.

## Differentiate between TCP and UDP.

Factor of comparison	TCP	UDP
Connection	Connection made before application messages are exchanged	Connection not made before application messages are exchanged
Use	For applications needing more reliability and less speed	For applications needing more speedy and less reliability
Use	File transfer, e-mail, etc	Multimedia, DNS
Reliability	Messages will be delivered in order and without errors	No guarantee that the messages will be delivered in order and without errors
Data Segments	Data segments rearranged in required order	All segments are independent, therefore has no inherent order specification
Acknowledgment	ACK is received	ACK is not received
Flow Control	Has the congestion control mechanism	No flow control option
Check for Errors	Resends erroneous segments	Discards Erroneous segments

## What is a MAC address and why is it required?

- ❖ MAC address is a computer's unique number NIC
- ❖ It is a 48-bit number
- ❖ Identifies each device on a network
- ❖ Referred to as the physical address
- ❖ Used as a network address for communications within a network

## What is piggybacking?

In two-way communication, the receiver sends an acknowledgment to the sender after receiving the data packet only after its network layer passes in the next data packet. The ACK then attached to the outgoing data frame. This process of is known as piggybacking.

## Explain what is DNS?

DNS is a naming system which is hierarchical and decentralized. It translates domain names to the numerical IP Addresses which is required to identify and locate devices based on the underlying protocols.

## What is Round Trip Time?

Round Trip Time or Round Trip Delay Time refers to the time taken for a signal to be sent and the ACK of that signal to be received.

## What is a Ping?

A ping is a computer program that is used to test the reachability of a host and check if it can accept requests on an IP network. It works by sending an ICMP Echo to some computer on the network and waits for a reply from it.

## What do you mean by Ethernet?

Ethernet

Network technology used in LAN, MAN and WAN that connects devices using cables for the transmission of data. It provides services on the Physical and Data Link layers of the OSI Model.

## What is an encoder?

An encoder is a program, circuit or a device that converts data from one format to another. Encoders convert analog signals into digital ones.

## What is a decoder?

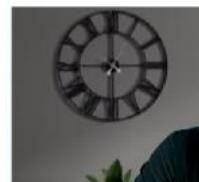
A decoder is a program, circuit or a device that converts the encoded data into its actual format. Decoders convert digital signals to analog ones.

**03**

### What is the purpose of ARP in a network?

tion:

The **Address Resolution Protocol (ARP)** is a network protocol used to **map an Internet Protocol address (IP address)** to a physical machine address that is recognized in the local network

**4**

### Discuss the different types of **OSPF areas** and their **functions**

on:

OSPF (**Open Shortest Path First**) is a dynamic **routing protocol** used within larger networks, known as **autonomous systems**. It utilizes a **link state routing (LSR) algorithm** to determine the best path for data through the network



**18**

What is a **denial-of-service attack**, and how can it be prevented?

solution:

A **denial-of-service (DoS) attack** is a common type of **cyber threat** that aims to make a website or network service unavailable to its intended users. The goal is to overwhelm the **target with a flood of useless traffic or requests**, which can cause the system to slow down significantly or crash, denying service to legitimate users



How do quality of **service (QoS) parameters** affect network **traffic management**?

on:

**Quality of Service (QoS)** is a technology used in networks to ensure that certain types of traffic get priority over others, **ensuring optimal performance** for **critical applications**, especially when the network is congested

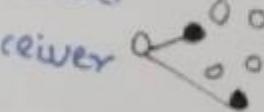


Explain Unicasting vs AnyCasting vs Multicasting  
vs Broadcasting

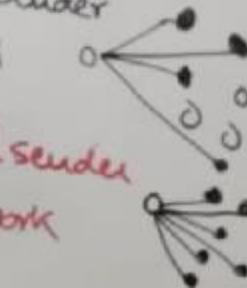
① Unicasting → 1 Sender, 1 receiver



② Anycasting → Data is sent from 1 sender to nearest or best receiver



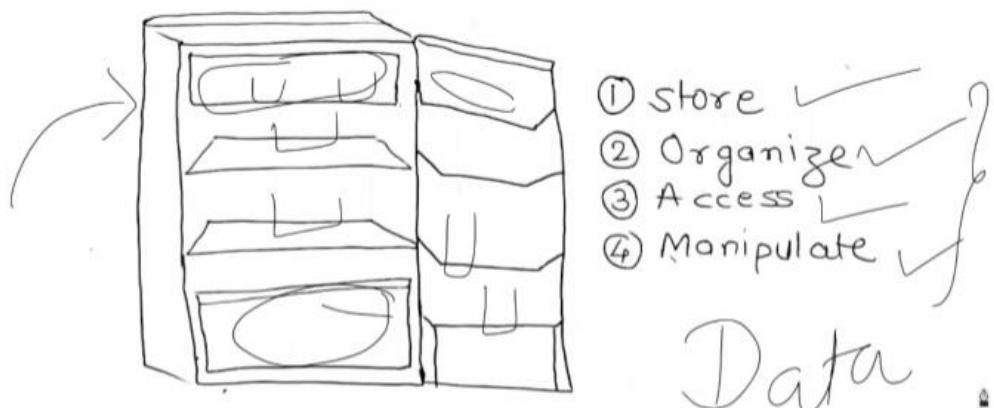
③ Multicasting → Data is sent from 1 sender to "group" of receiver interested



④ Broadcasting → Data is sent from 1 sender to "All" receivers in a Network



① What is Data structure?



②

LDS

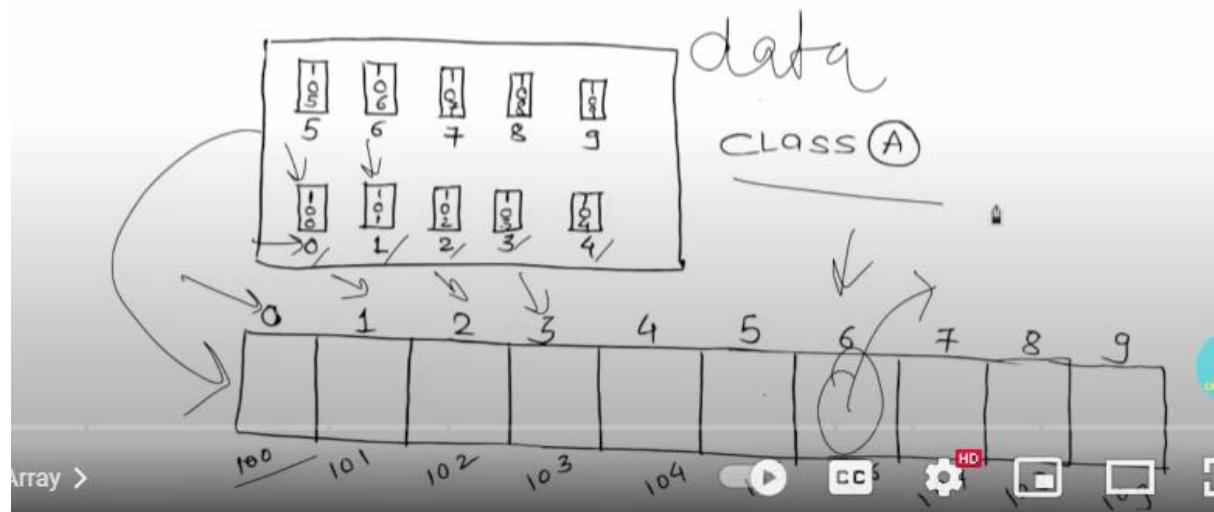
DDDS

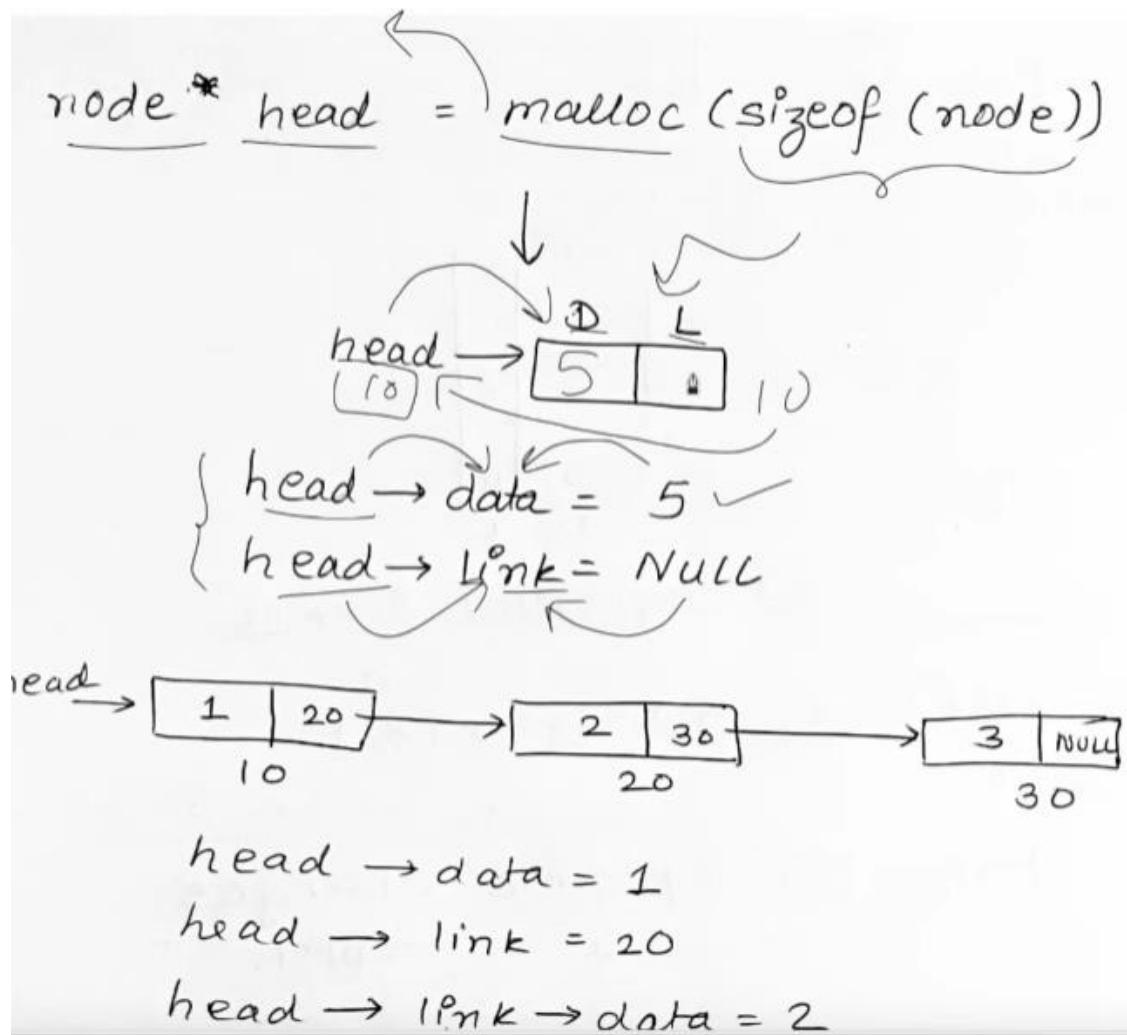
NLDS

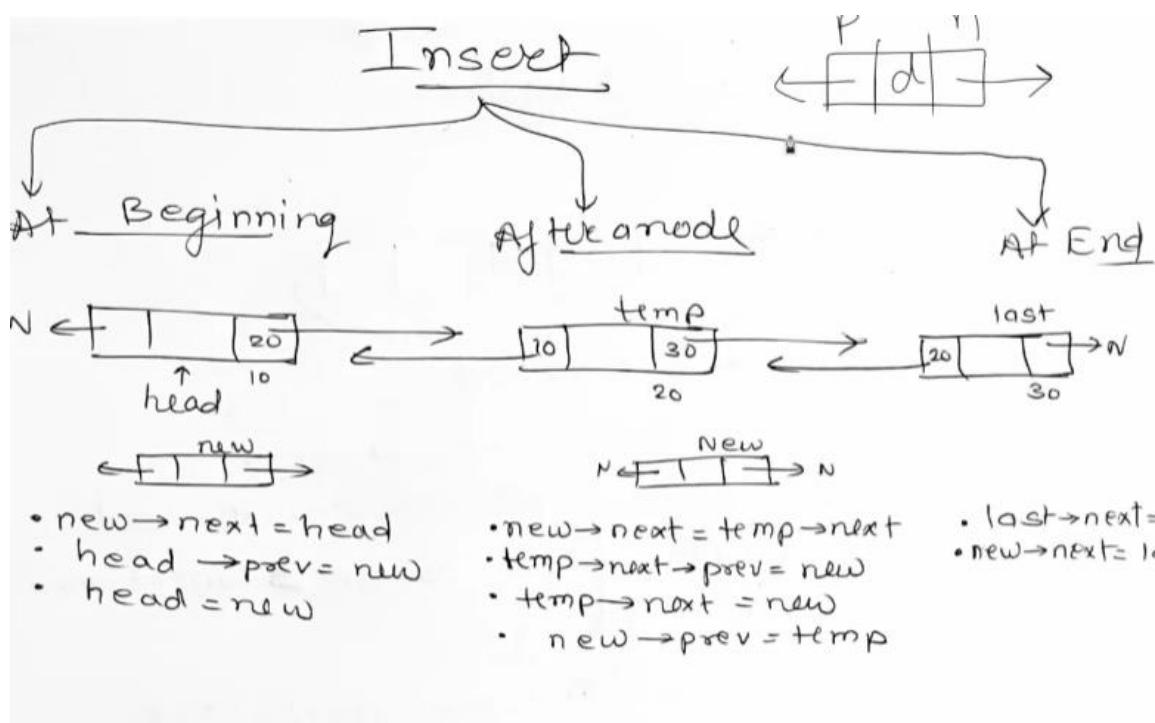
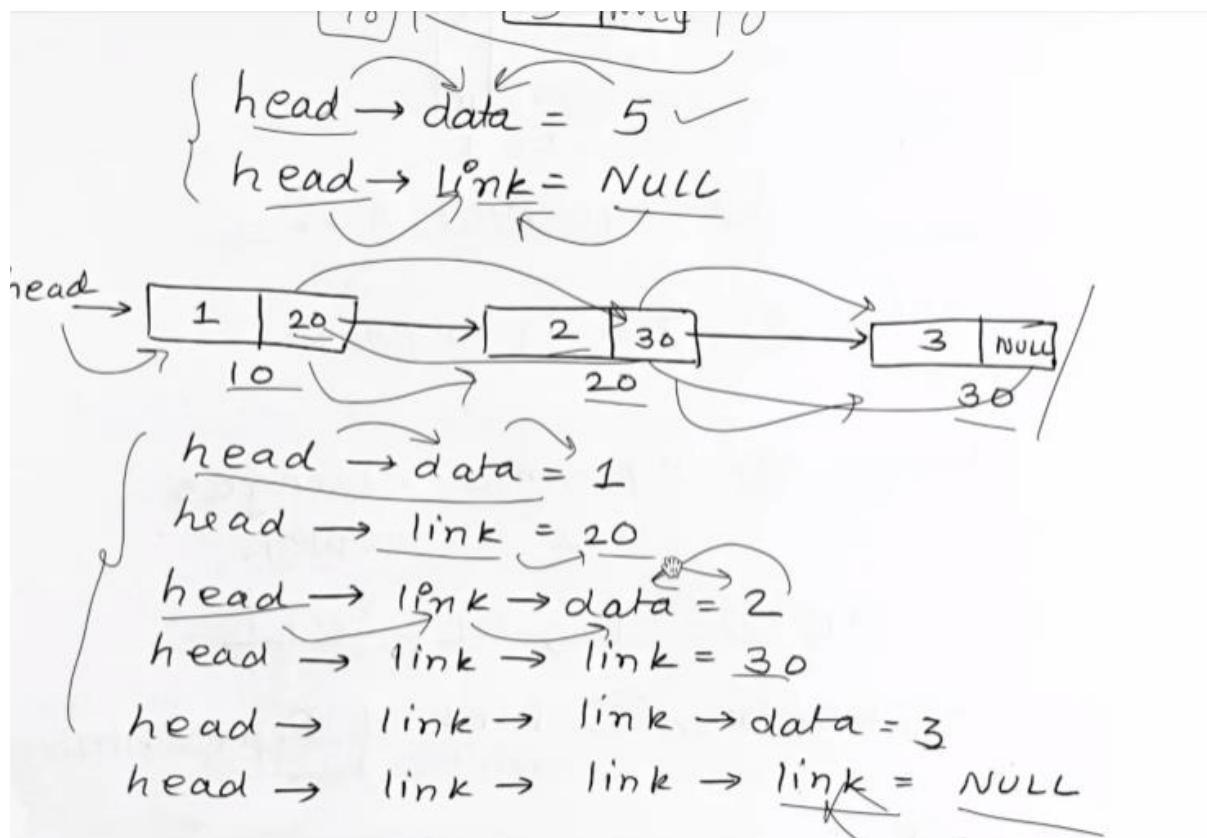
- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>→ <u>Adjacently Attached</u></li> <li>→ <u>single Level</u></li> <li>→ <u>Easy</u></li> <li>→ <u>Single Run</u></li> <li>→ <u>Memory is not Utilized in efficient way</u></li> <li>→ <u>Array, stack, LL</u></li> </ul> | <ul style="list-style-type: none"> <li>→ <u>Hierarchically Attached</u></li> <li>→ <u>Multiple Level</u></li> <li>→ <u>Difficult</u></li> <li>→ <u>Multiple Runs</u></li> <li>→ <u>In efficient way</u></li> <li>→ <u>Tree Graph</u></li> </ul> |
|--|---|

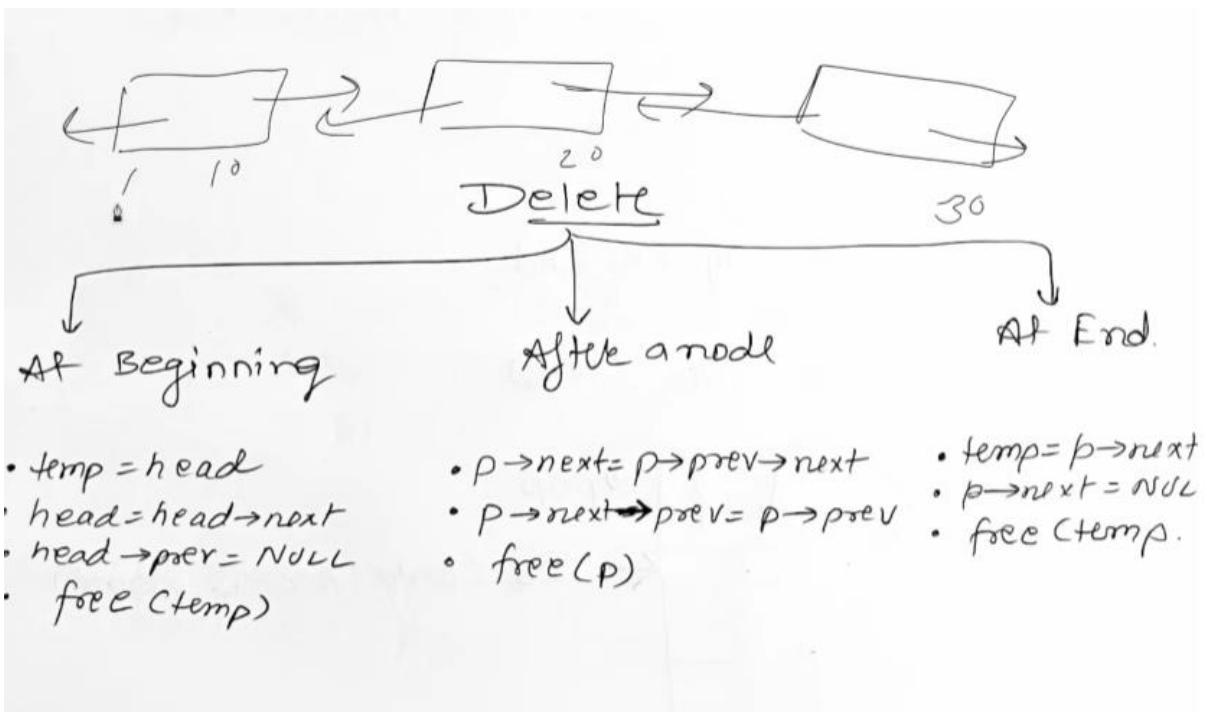
### 3) Array

- store ✓
- same type ✓
- contiguous memory locations }  
→ index accessing ✓

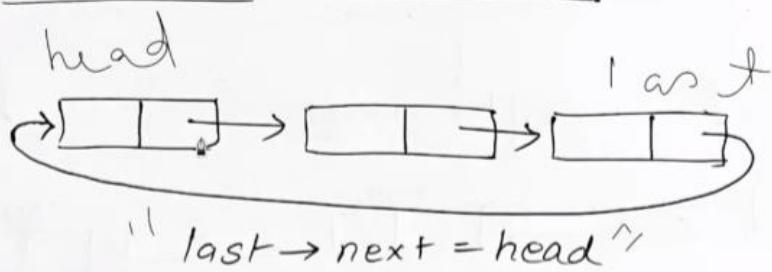




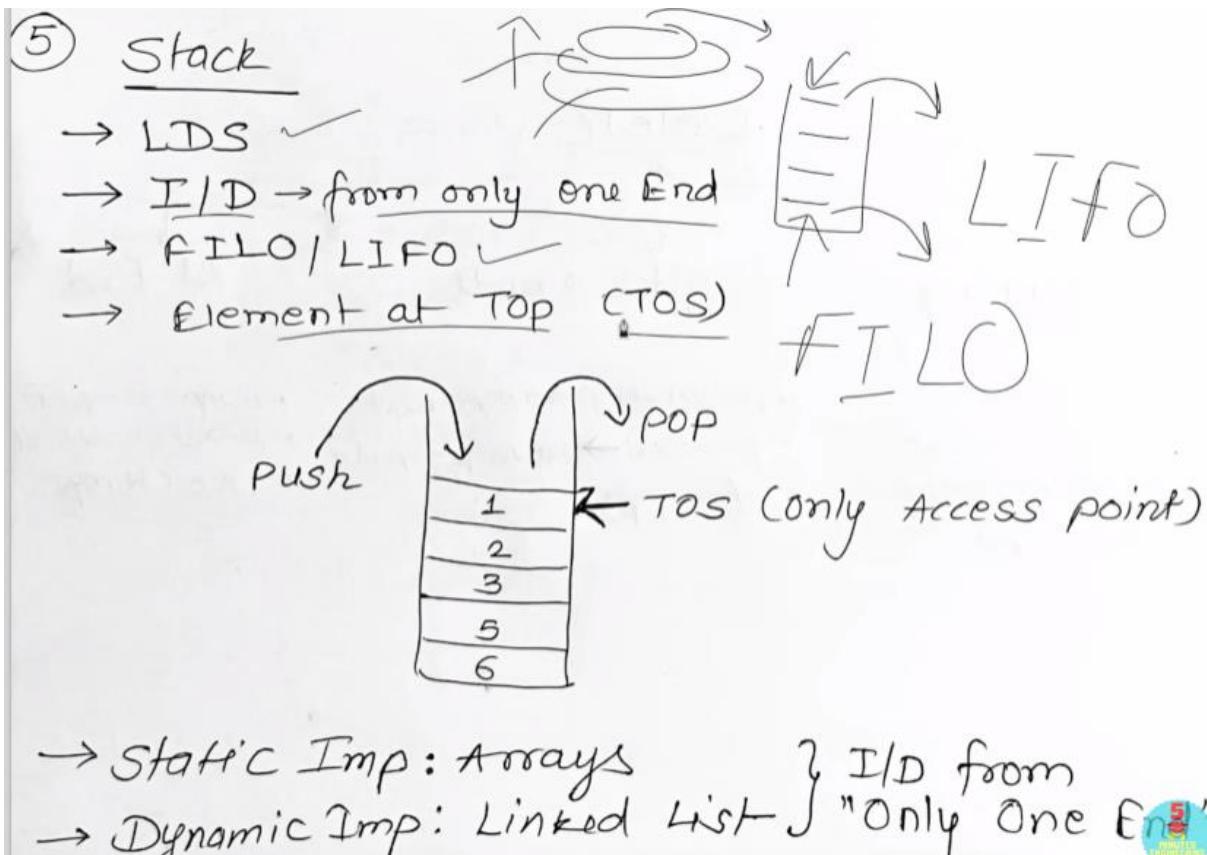




### → Circular Linked List



- ② •  $\text{temp} = \text{head}$
  - ③ •  $\text{head} = \text{head} \rightarrow \text{next}$
  - ⑤ •  $\text{free}(\text{temp})$
- current = head*  
 while ( $\text{current} \rightarrow \text{next} \neq \text{head}$ )  
 {  
     *current*  $\Rightarrow$   $\text{current} \rightarrow \text{next}$   
 }  
 ④  $\text{current} \rightarrow \text{next} = \text{head}$



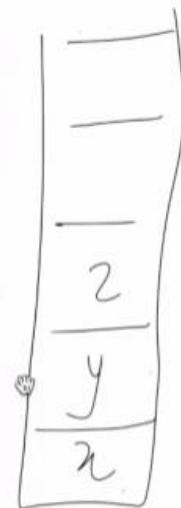
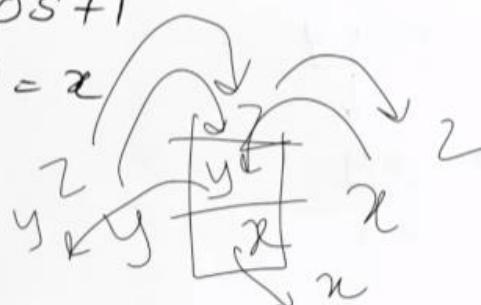
→ Push (arr, n, TOS, x)

{  
If (TOS == n - 1)  
{ printf("stack is full")  
}

TOS = TOS + 1

arr [TOS] = x

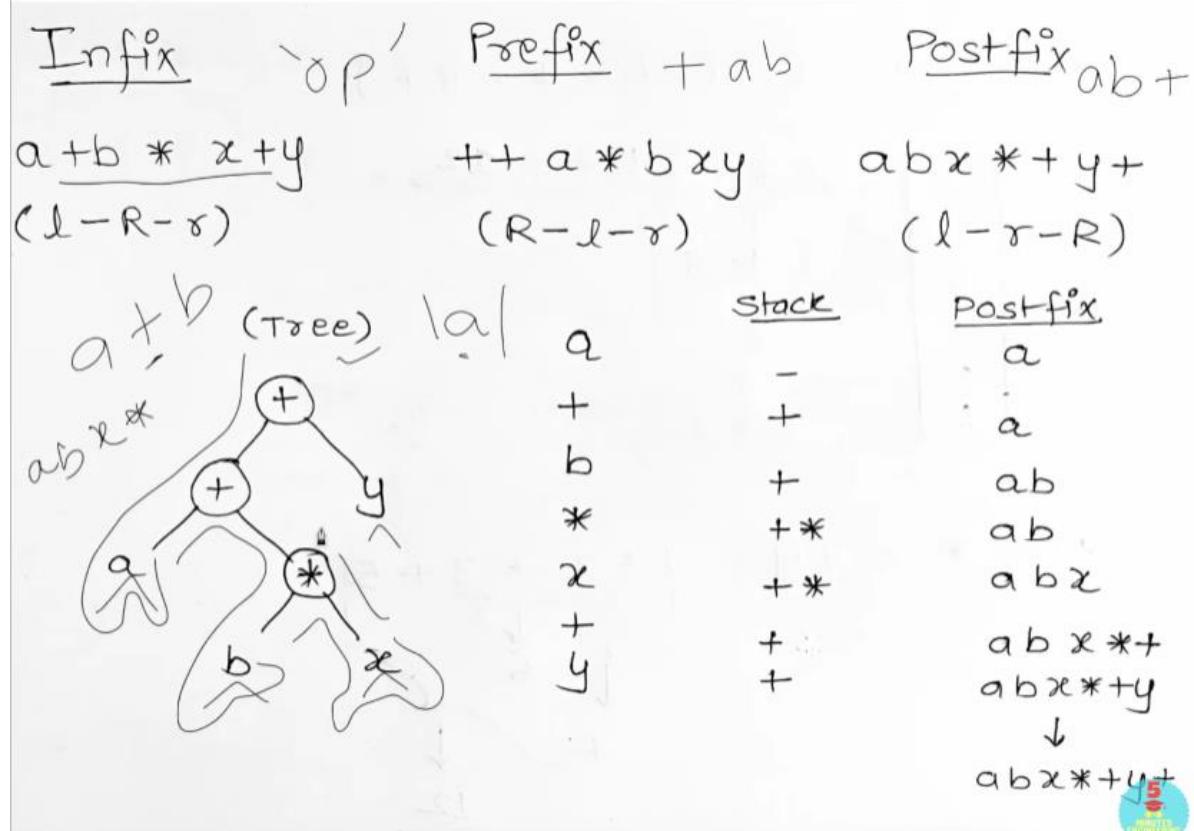
}



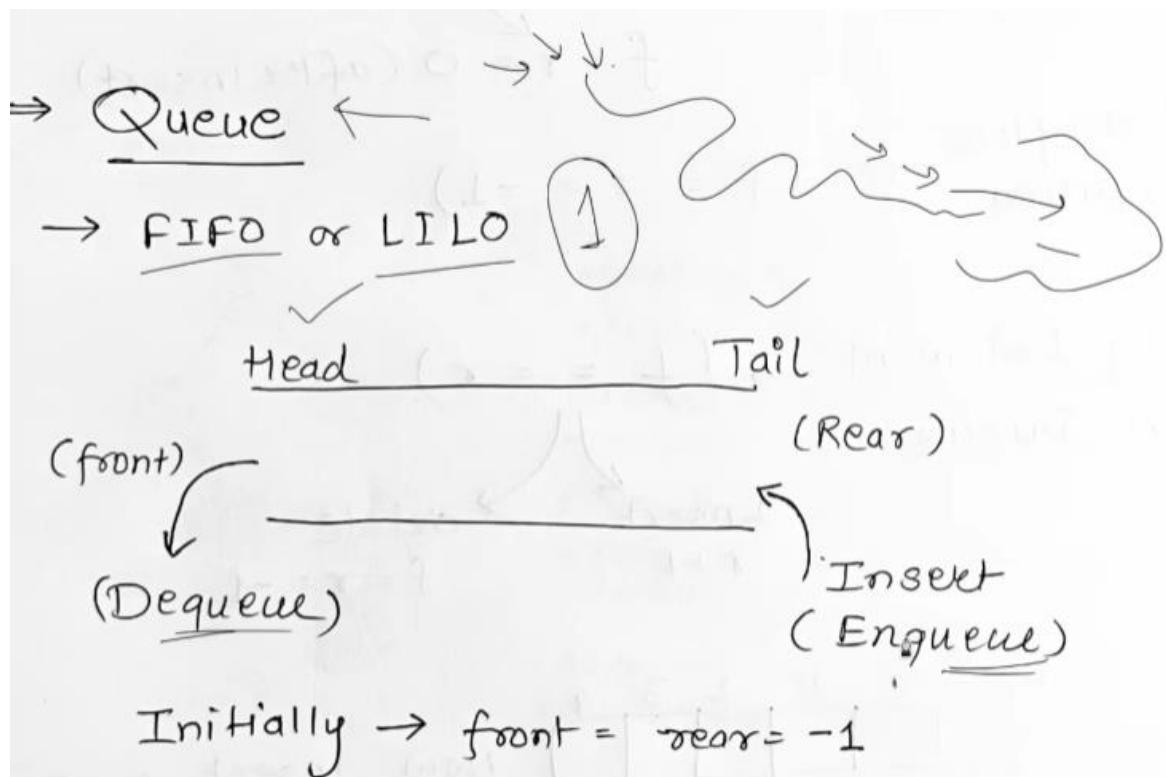
→ Pop (arr, n, TOS)

{  
If (TOS == -1)

{ printf("stack is empty")



Infix – 2<sup>nd</sup> occurrence  
prefix – 1<sup>st</sup> occurrence  
postfix – last occurrence

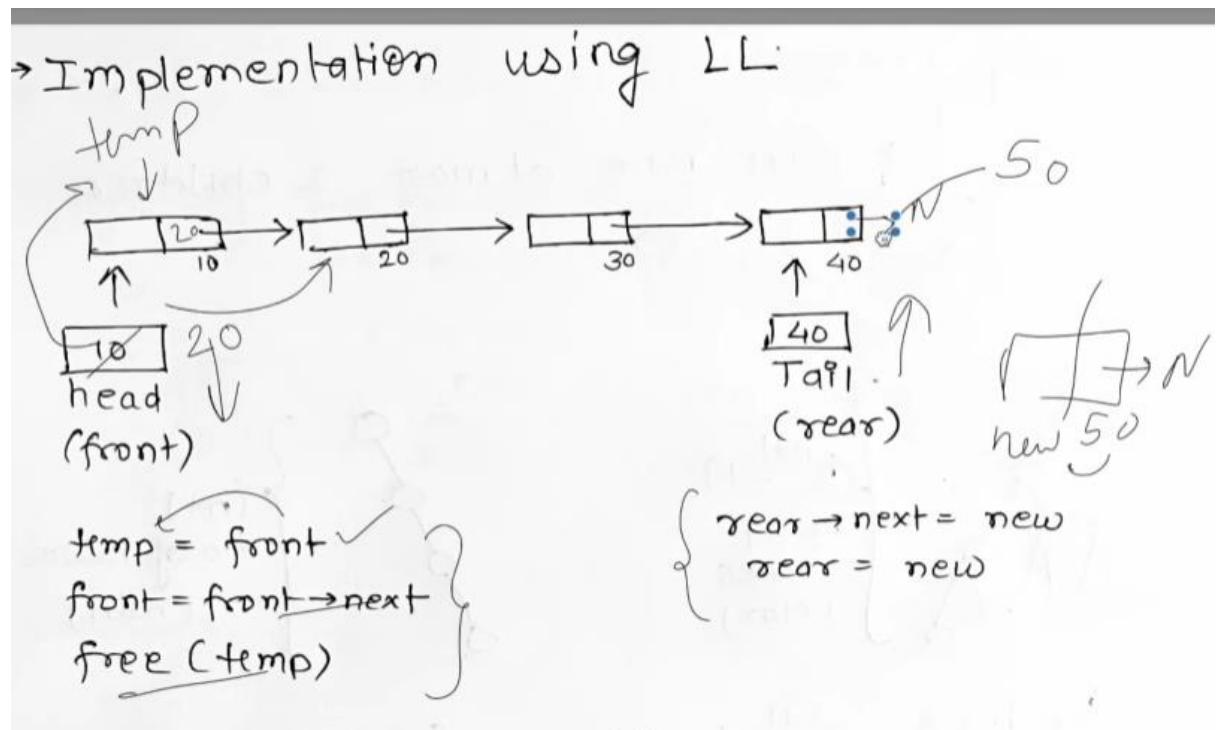


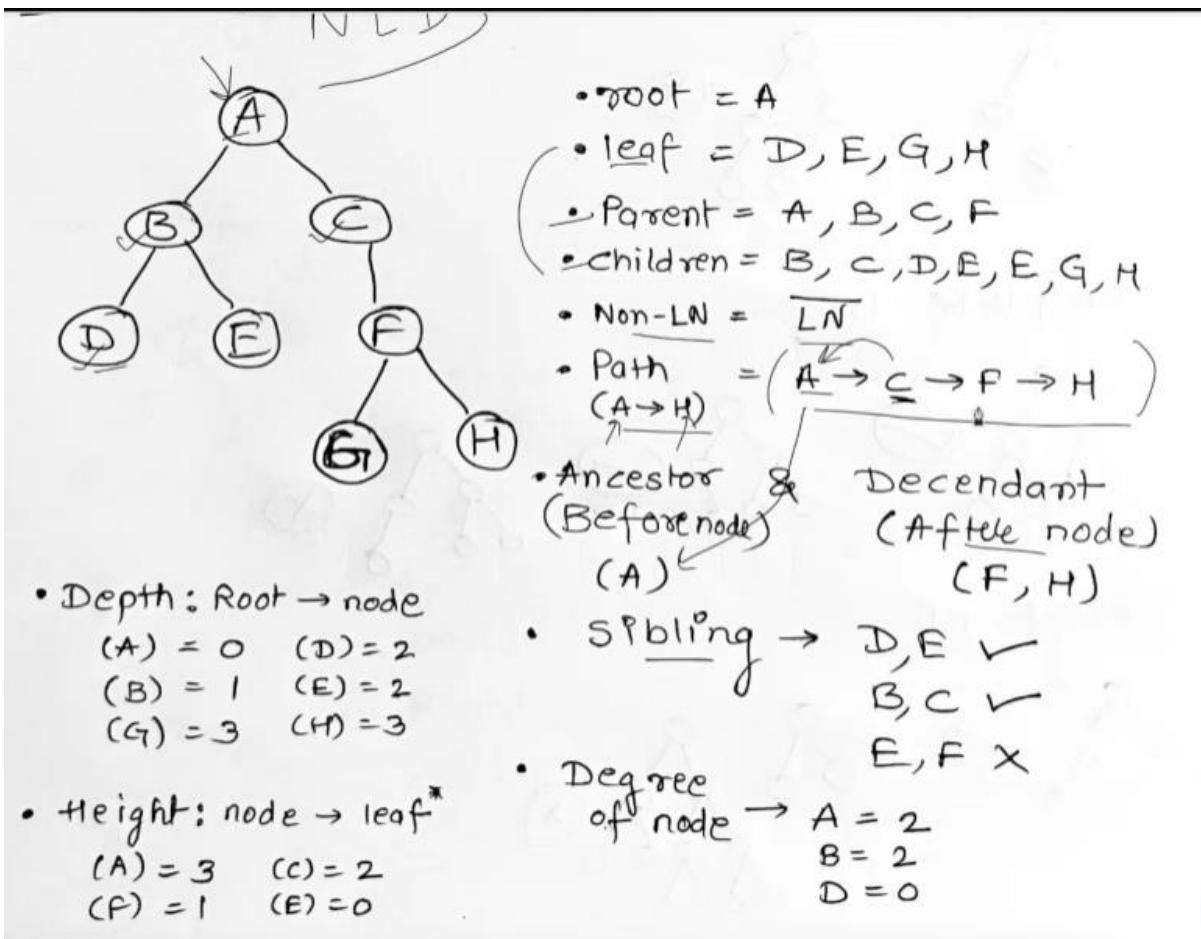
### Implement a stack using queues:

- **Solution:** Use two queues. Push to the first queue, and during pop, transfer all elements to the second queue, leaving one for the pop operation.

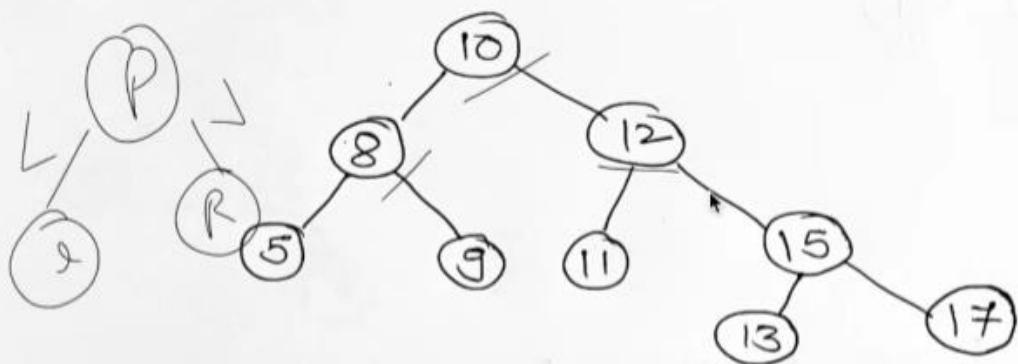
### Implement a queue using stacks:

- **Solution:** Use two stacks. For enqueue, push to the first stack; for dequeue, transfer elements from the first stack to the second stack and pop.



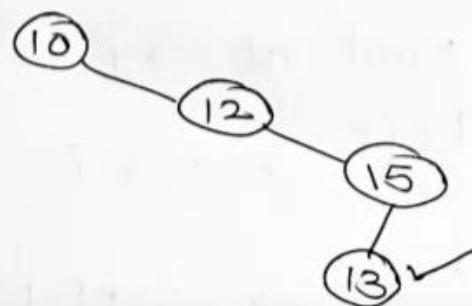


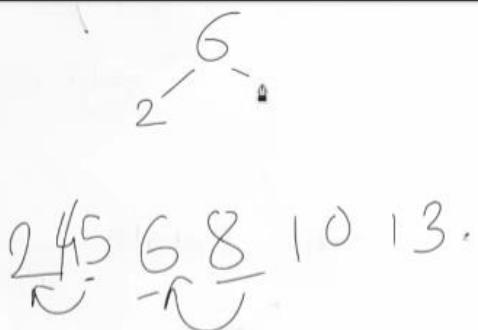
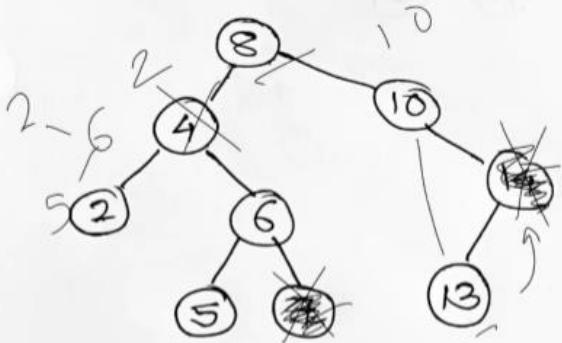
→ BST



Inorder  $\Rightarrow \{5, 8, 9, 10, 11, 12, 13, 15, 17\}$

- Searching becomes easy (13)



Deletion

- leaf node (Easy) direct (NO Impact)
- One child node (4) → Replace it with that child (13)
- 2 child node (4) → Replace it with inorder predecessor. (2)

2, 4, 5, 6, 7, 8, 10, 14, 13

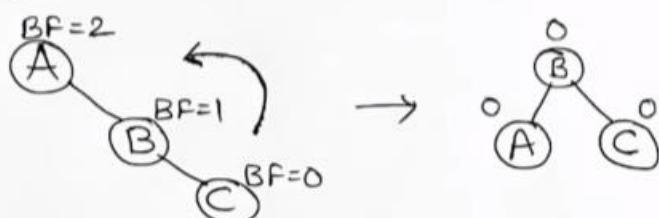
AVL Tree

↳ Gets a self-balancing BST

$$Bf = H(LST(n)) - H(RST(n))$$

↳  $\{-1, 0, +1\}$

## ① Left Rotate

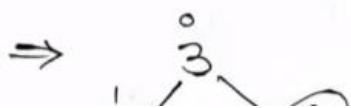
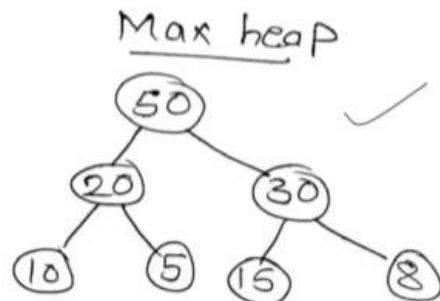
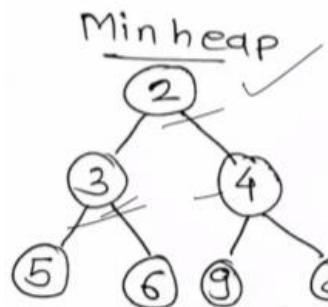


→ Heap / L → R

→ Complete Binary Tree ✓

    └ Min heap ( Parent < child )

    └ Max heap ( Parent > child )



(vertices)

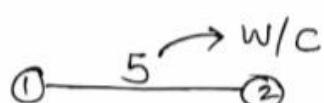
① Directed Graph



② Undirected Graph



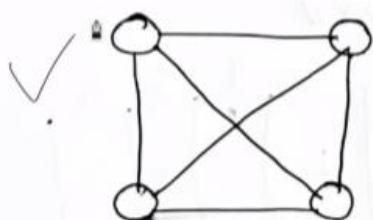
③ weighted Graph



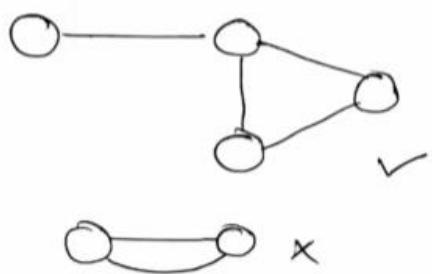
④ Unweighted Graph



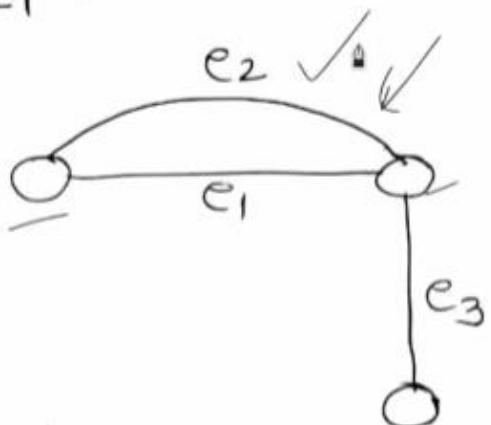
⑤ Complete Graph



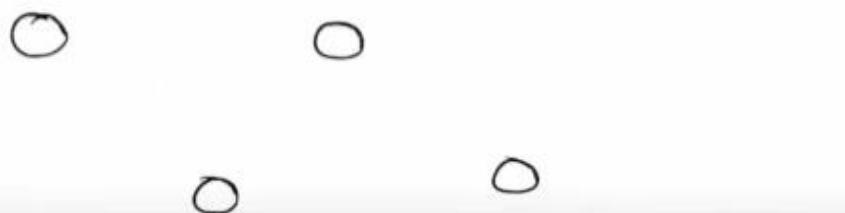
⑥ simple Graph



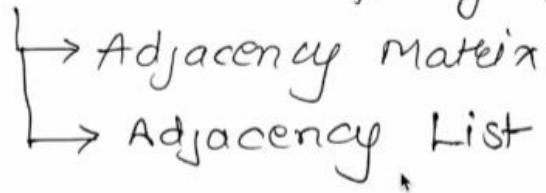
⑦ Multi Graph



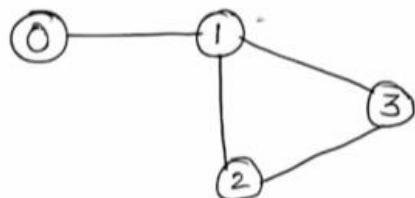
⑧ NULL Graph



⇒ Representation of graph:



Eg:



$$V = 4$$

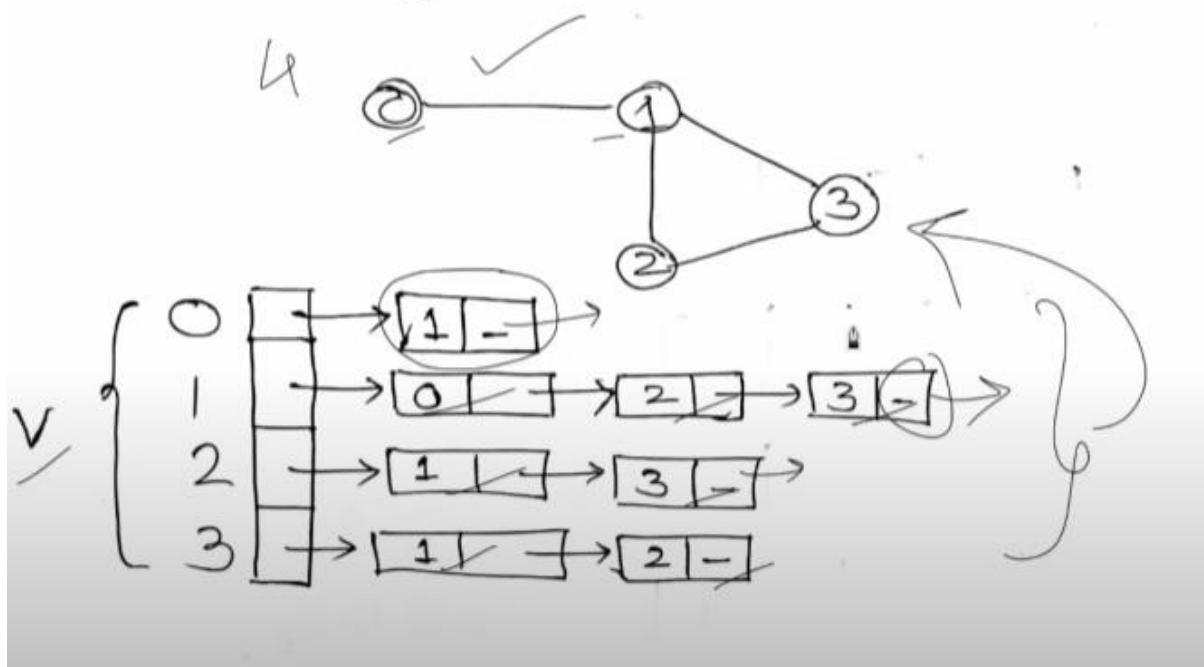
$$E = 4$$

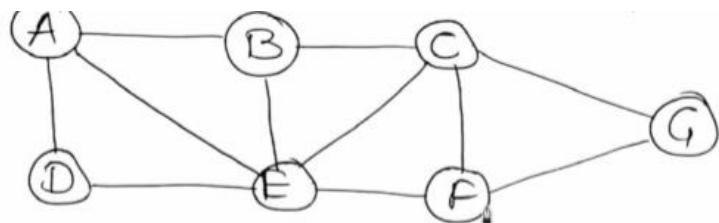
$$\begin{matrix} 1 & \rightarrow & \checkmark \\ 0 & \rightarrow & \times \end{matrix}$$

		0	1	2	3
0	0	1			
	1		1	1	
2			1		
3				1	1

Matrix m space bht waste ho rhi thi to adjancy list le aaye hum

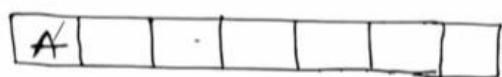
⇒ Adjacency List



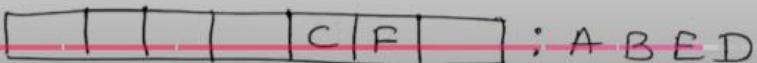
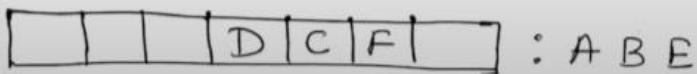
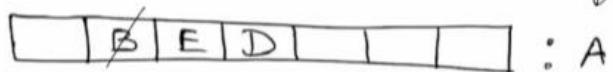


✓ Start with (A)

⇒



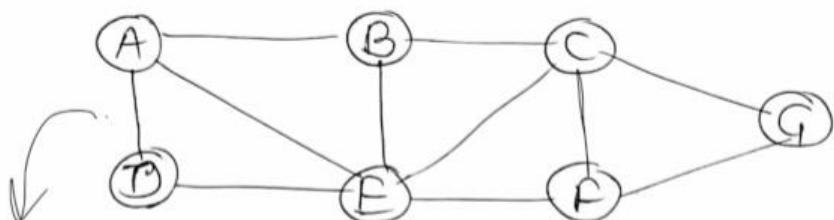
⇒



Graph >



## DFS



- $A \rightarrow \underline{B, D, E}$
- $B \rightarrow A, C, E$
- $C \rightarrow B, E, F, G$
- $D \rightarrow A, E$
- $E \rightarrow A, B, C, D, F$
- $F \rightarrow C, E, G$
- $G \rightarrow C, F$

"A B C D E F G"

Ans: A E F G

Miniplayer (i)

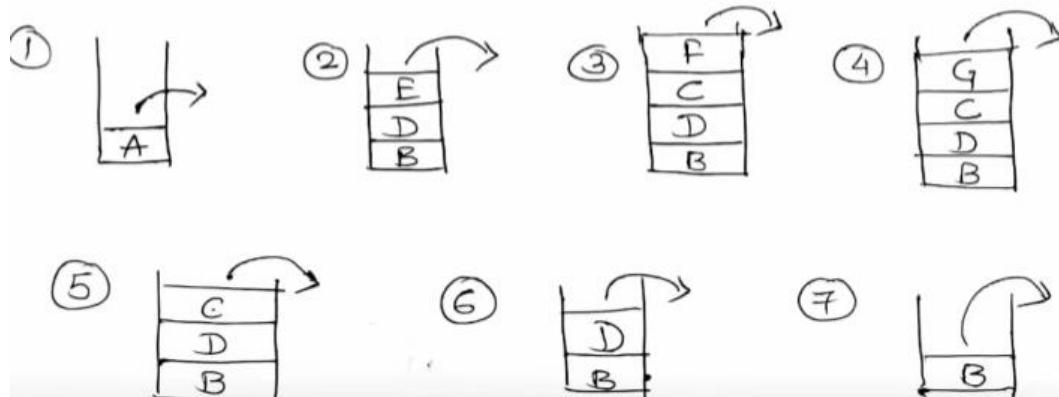
Graph >



$A \rightarrow \underline{B, D, E}$
$B \rightarrow A, C, E$
$C \rightarrow B, E, F, G$
$D \rightarrow A, E$
$E \rightarrow A, B, C, D, F$
$F \rightarrow C, E, G$
$G \rightarrow C, F$

"A B C D E F G"

Ans: A E F G C D B



→ Hashing → 'S' fast

(Time ↓)

Space ↑)

→ Key ✓

→ Location ✓

→ Hash  $f^n$  ✓

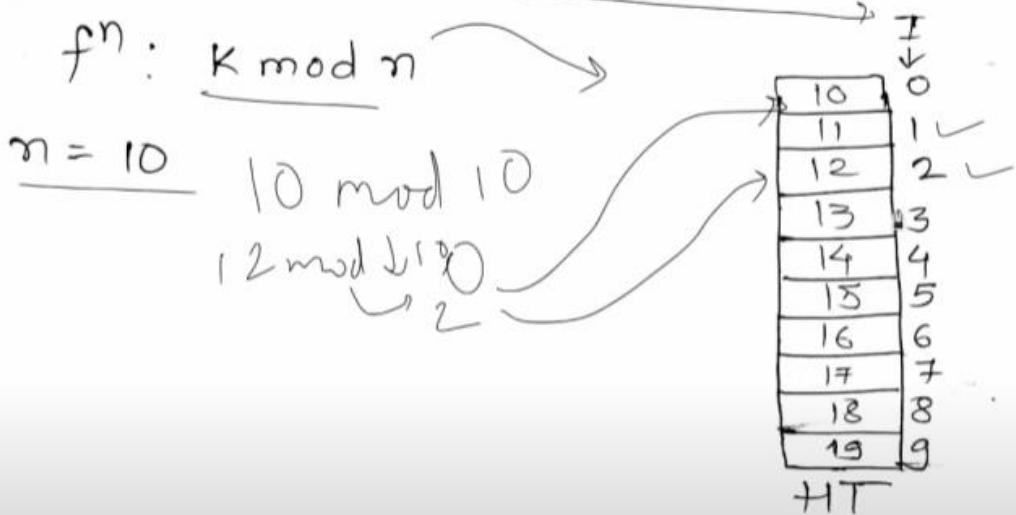
Easy  
Fast  
Low

→ Hash table. ✓

eg:  $f^n(H(k)) = k \bmod n$



Keys  $\rightarrow (10, 11, 12, 13, 14, 15, 16, 17, 18, 19)$



HT

• Mid Square ✓

$$\Rightarrow \underline{\underline{12}} \rightarrow \text{Square} \rightarrow \underline{\underline{144}}$$

$$\begin{matrix} & I=4 \\ \downarrow & \\ \text{store } 12 \end{matrix}$$

$$\Rightarrow \underline{\underline{11}} \rightarrow \text{Square} \rightarrow \underline{\underline{121}}$$

$$\begin{matrix} & I=2 \\ \downarrow & \\ \text{store } 11 \end{matrix}$$

•  $k = 123456$  (folding method)

$$\begin{matrix} & \swarrow & \searrow \\ 123 & + & 456 \\ \downarrow & & \end{matrix}$$



$k = \frac{123}{\cancel{4}} \quad \text{(folding method)}$

$$123 + 456$$



$$\begin{array}{r} 123 \\ + 456 \\ \hline \end{array} \quad \rightarrow 579 \quad \begin{matrix} \downarrow \\ (\text{mod } n) \end{matrix} \quad \begin{matrix} \downarrow \\ 10 \end{matrix}$$

$\therefore n = 10$ , so  $579 \bmod 10$

$\Rightarrow 9$  (store 123456)

#1
IntelliPaa



Handling



Storage

**What are Data Structures?**

- Methods and techniques used to maintain data in an organized fashion
- Define **data dependency** and relationships

**What is the difference between a File Structure and a Data Structure?**



File Structure	Data Structure
Data stored on disk	Data stored both on RAM and disk
Standard file storage policies	Customized storage policies
Low compatibility with external apps	High compatibility with external apps

### What is a **Linked List**?

- **Data structure** that consists of individual entities called **nodes**
- Nodes have the capability to connect to other nodes and create a chain
- The continuous chain structure forms to become a linked list

### Where are **Data Structures** primarily used?

- Numerical computation
- Operating system design
- Artificial Intelligence
- Compiler design
- Database handling
- Graphical processing
- Lexical analysis
- Statistics

### What are the types of **searching** used in Data Structures?

- **Linear search** involves iterating over a data unit in order to perform the required operation.
- **Binary search** is more efficient in a way that it has the ability to split the data unit into chunks and then perform a search operation.

### How does **Binary Search** work?

- **Binary search** works on ordered data (sorted) only
- To begin with, the middle element of the array is found out, and the search begins from there.
- The array is searched in **two parts** based on the search value being higher or lower than the middle element.
- It is key to know the order of the arrangement to help search the value accordingly.

### How are individual elements accessed in an **array**?

- Each of the values in an array is given an index position starting from **0 to n-1**, where 'n' is the number of elements in the array.
- Individual elements can be accessed by using the index element for operations.
- Multi-dimensional arrays will have **more than one dimension** to work with.

## What is a **queue** in Data Structures?

- A **queue** is a widely used data structure that is used to denote the ordered access and manipulation of an element.
- The operation of this data structure is exactly the **same** as a literal queue in the real world.
- Elements are added one after the other and are processed on the front end.

## What is a **Binary Tree**?

- A **binary tree**, as the name suggests, is a tree data structure with **two nodes**, which are the nodes on the left and the right sides of the root node.
- In usage, binary trees are considered to be an **extended linked list**.

## What is the meaning of **stack**?

- A **stack** is another widely used data structure that provides users with the ability to work with data at one point only.
- As the name suggests, this can literally correspond to the working of a **stack of cards**.

## What is the working of **LIFO**?

- **LIFO** stands for the **Last in, First out** access order.
- Corresponds to how the data can be worked on and modified.
- The data entity that is stored or pushed in last is the first one to be worked on at any point in time.
- If there is a requirement to access the very first element stored, then first you have to retrieve all of the data that came in after that element.

## What are **multi-dimensional arrays**?

- **Multi-dimensional arrays** are arrays that span across more than one dimension.
- There will be more than one index variable for every point of storage.
- Used in cases where data **cannot** be represented or stored using only one dimension.

like gps coordinates for example

## Are linked lists **Linear** or **Non-linear** Data Structures?

- **Linked lists** are considered to be the best of both worlds here.
- Based on usage, if it is a storage policy, then it can be considered as non-linear.
- Whereas, if a person is considering it based on retrieval strategies, then it can be considered **linear**.

### What is a Binary Search Tree?

- It consists of **two** primary nodes from the root node.
- Values of the nodes in the left subtree are less in number than the value of the root node, and the values of the nodes on the right of the root node are correspondingly higher than the root.
- Also, individually both of these left and right subtrees are their own binary search trees at all points of time.

### What is the meaning of FIFO?

- **FIFO**, also known as First in, First out, is a way of representing a data operation on factors such as how data is accessed and in what order.
- Here, the data that is first put into the list will be the first entity to exit from the ordered data structure.

### What is the difference between void and null in Data Structures?

- **Void** is a data type identifier in data structures, while **null** is considered to be a value with no physical presence.
- When **void** is used, it indicates that there is no size while initializing the data structure.

### What is dynamic memory management?

- Dynamic memory management is a technique in which storage units are allocated based on the requirements continuously.
- Using dynamic memory allocations, individual data structures can be either stored separately or combined to form entities called **composites**.
- These composites can be worked on when required.

### What are push and pop operations in Data Structures?

- The **push** operation denotes that users are adding data into the structure.
- **Pop** operation denotes that the data is being pulled or removed from the structure.
- Usually, the top-most element is considered when performing push and pop operations.

### How is a variable stored in memory when using Data Structures?

- A **variable** is stored based on the amount of memory that is needed.
- First, the required quantity of memory is **assigned**, and later, it is stored based on the data structure being used.
- Using concepts such as dynamic allocation ensures high efficiency and that the storage units can be supplied based on the requirements in real time.

### What is merge sort?

- Merge sort is a method of sorting, which is based on the divide and conquer technique.
- Here, data entities adjacent to each other are first merged and sorted in every iteration to create sorted lists.
- These smaller sorted lists are combined at the end to form the completely sorted list.

### Why should heap be used over a stack?

- The heap data structure is more efficient to work with when compared to stack
- Because the memory allocation in a heap is dynamic and can be allocated and removed as per the requirement.
- The memory structure and access time of a stack are comparatively slow.

### What is the meaning of Data Abstraction?

- Data abstraction is one of the widely used tools in data structures.
- The goal is to break down complex entities into smaller problems and solve these by using the concepts of data structures.
- This provides users with the advantage of being focused on the operations and not worried about how the data is stored or represented in the memory.

### What is the meaning of a postfix expression in Data Structures?

- Postfix expressions are used in a scenario where every operator is preceded by its operands.
- Eliminates the need for parentheses or subexpressions when it comes to the concept of operator precedence.

### What is the working of a selection sort?

- The working is simple where the smallest entity is first found out and the index of that is set to zero, thereby permanently sorting this in the first step.
- The remaining steps involve iterating through other elements and adding the next index correspondingly. This is done until all of the elements are sorted.

### What are signed numbers in Data Structures?

- Signed numbers are the units that have a data bit at the beginning of the number that denotes if the number is positive or negative.
- Signed numbers have a range of -128 to +127.

1 – negative

0 – positive

**What are the minimum nodes binary trees can have?**

- Binary trees can have **zero** nodes or a minimum of 1 or 2 as well.
- It can be zero in a case where all of the nodes have a **NULL** value.

**What Data Structures make use of pointers?**

Pointers are used in a variety of data structures. They are majorly used in the following data structures:

- Binary trees
- Linked lists
- Queues
- Stacks

**What is the use of dynamic Data Structures?**

- **Dynamic data structures** provide users with a lot of flexibility in terms of the provision of data storage and manipulation techniques
- Used to change during the operation of the algorithm or the execution of the program.

**What is a priority queue?**

- A **priority queue** is an abstract data type that resembles a normal queue but has a priority assigned to elements.
- Elements with higher priority are processed before the elements with a lower priority.
- A minimum of two queues are required in this case, one for the data and the other to store the priority.

ambulance waala example

**Pointers allocate memory for data storage. True or False?**

- **False**, pointer operations such as declaration will not allocate any memory for the storage of data.
- But, memory is allocated for the variable that the pointer is pointing to.
- Memory processing begins only when the program begins its execution.

**What is the meaning of deque?**

- A **deque** is a queue that is double-ended.
- This means that elements can be added or removed from any one of the two ends. It can be used both as a regular queue and as a stack.
- It performs better than both linked lists and stacks in general.

## Differentiate between Linear and Non-linear Data Structures

Linear Data Structures	Non-Linear Data Structures
Data elements are stored next to each other	Data elements can be separated by other entities in memory
E.g.: Arrays, linked lists, stacks, and queues	E.g.: Trees and graphs

## What is the meaning of an AVL tree?

- An AVL tree is a type of a binary search tree where the tree is only slightly balanced.
- Balance is the unit of comparison between the heights of the subtrees from the main (root) node.

## How does Huffman's algorithm work?

- Huffman's algorithm uses a table, containing the frequency of the occurrence of every data entity on the list.
- This is used for creating extended binary trees, which are known to have minimum weights for the path lengths .
- This is considering each of the corresponding weights.

## What are recursive algorithms?

- Recursive algorithms are algorithms that solve a problem by breaking it down into simpler sub-problems and then solving them iteratively.
- The output of one recursion operation is usually the direct input for the next iteration operation, and this process goes on.

## How does bubble sort work?

- Applied to arrays where elements adjacent to each other are compared and values are exchanged based on the order of arrangement.
- It's called bubble sort because of the concept of exchanging elements like a bubble floating to the top of the water and larger entities sinking down to the bottom end.

## Which is the fastest sorting algorithm available?

- Among the many types of algorithms such as bubble sort, quick sort, merge sort, and more, it is not right to put one method on the podium for performance
- As this greatly varies based on data, the reaction after the algorithm processes the data, and how it's stored.
- The concept of time complexity is considered here.

**What is the postfix form of:**  
 $(X + Y) * (Z - C)$

The postfix form of the given expression is **XY+ZC-\***

**Where are Tree Data Structures used?**

**Tree data structures** are used in a variety of applications. Following are some of them:

- Arithmetic expression handling
- Symbol table creation
- Lexical analysis
- Hierarchical data modeling

**What are the Data Structures that are used in graphs?**

To implement graphs, two data structures play a key role. They are:

- **Adjacency matrix:** Used for sequential data representation
- **Adjacency list:** Used to represent linked data

**What are the Data Structures that are used in DFS and BFS algorithms?**

- In the **depth-first search (DFS)**, the stack data structure is made use of.
- In the case of the **breadth-first search (BFS)** technique, queues are used.

**What are the time complexities of linear search and binary search?**

- **Binary search** is more effective as it takes lesser comparisons to search for an element in an array.
- The time complexity for **linear search** is  $O(n)$
- Time complexity is  $O(\log n)$  for **binary search**.

**Where are Multi-linked Data Structures used?**

**Multi-linked data structures** are used in many domains. Following are the two most important use cases of multi-linked data structures:

- Generation of sparse matrices
- Index creation for data entities

**What is the method used for inorder traversal in trees?**

**Inorder traversal** works in the following way:

- The tree is traversed through the left subtree.
- The root node is visited after the left visit.
- Then, the right subtree is traversed.

**What is the working of postorder traversal in trees?**

**Postorder traversal** works in the following way:

- First, the left subtree is traversed through.
- The right subtree is traversed next.
- The root node is visited after the right subtree visit.

**What are the disadvantages of implementing queues using arrays?**

- **Array sizing:** The queue has to be constantly extended to make way for more elements that get implemented. Always extending the size of the array will not be feasible as there will be a discrepancy in the creation of the correct array size.
- **Memory dumps:** The memory that is used to store the queue elements cannot be reused to actually store the queue. This is because of the working of queues where insertion happens at the head node only.

**How can elements be inserted in the circular queue?**

There are **two** cases in which items can be placed in a circular queue. They are as follows:

- When **front != 0 and rear = max -1**. This makes it possible as the queue will not be full, and new elements can be inserted here.
- When **rear != max -1**. This ensures that the rear is incremented to the maximum allocation size, and values can be inserted easily to the rear end of the queue.

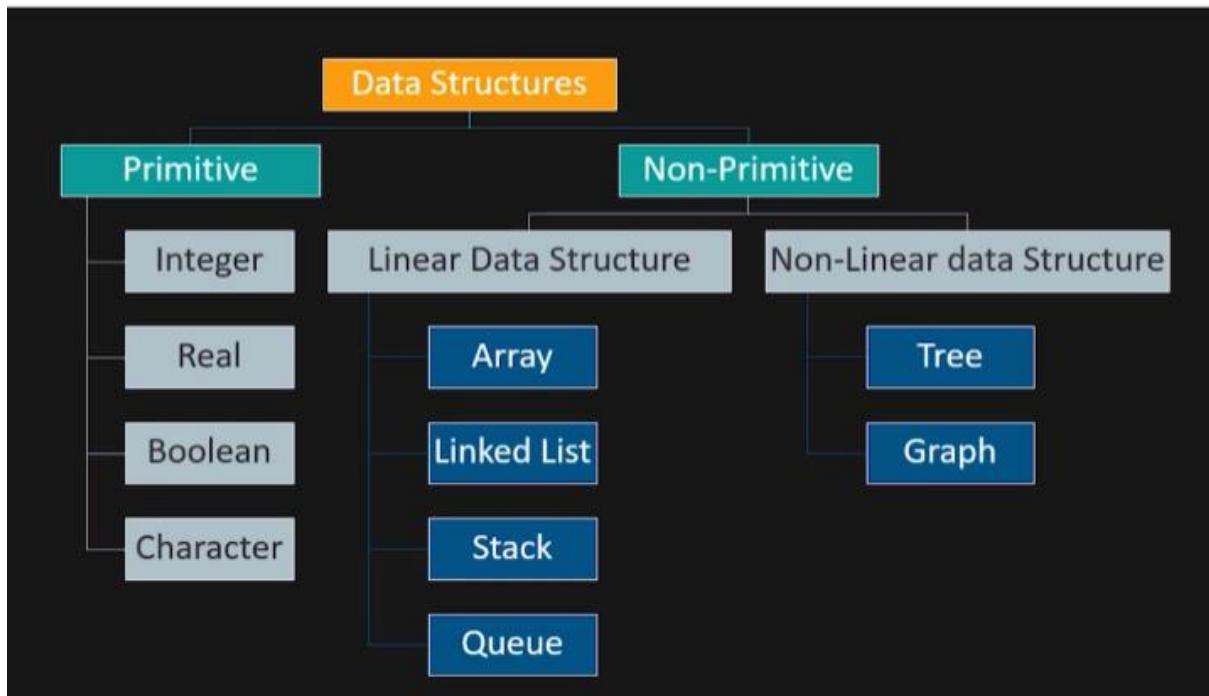
**What is the use of void pointers?**

- **Void pointers** are used because of their capability to store any pointer, which is pointing to a wide variety of data.
- It is used to implement heterogeneous linked lists in many programming languages.

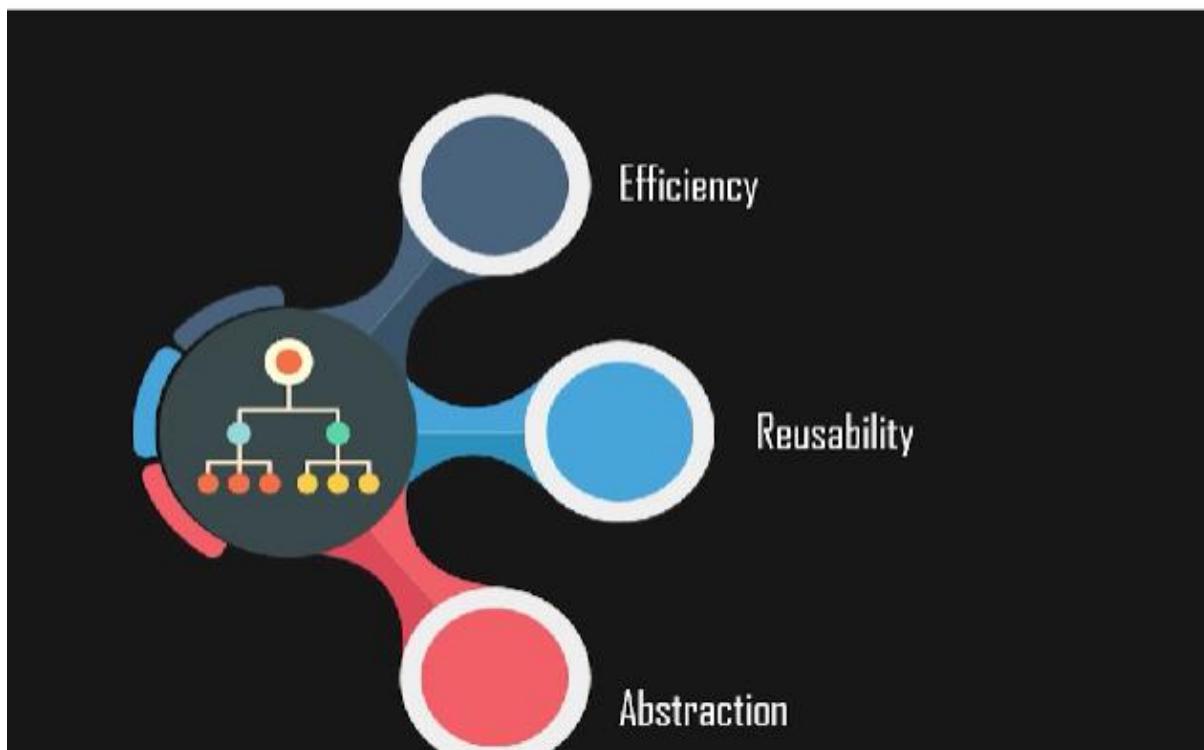
What is the meaning of the stack overflow condition?

- Stack overflow is the term given when the stack is full and an element cannot be inserted into the stack anymore.
- Stack overflow happens when `top = Maxsize - 1`

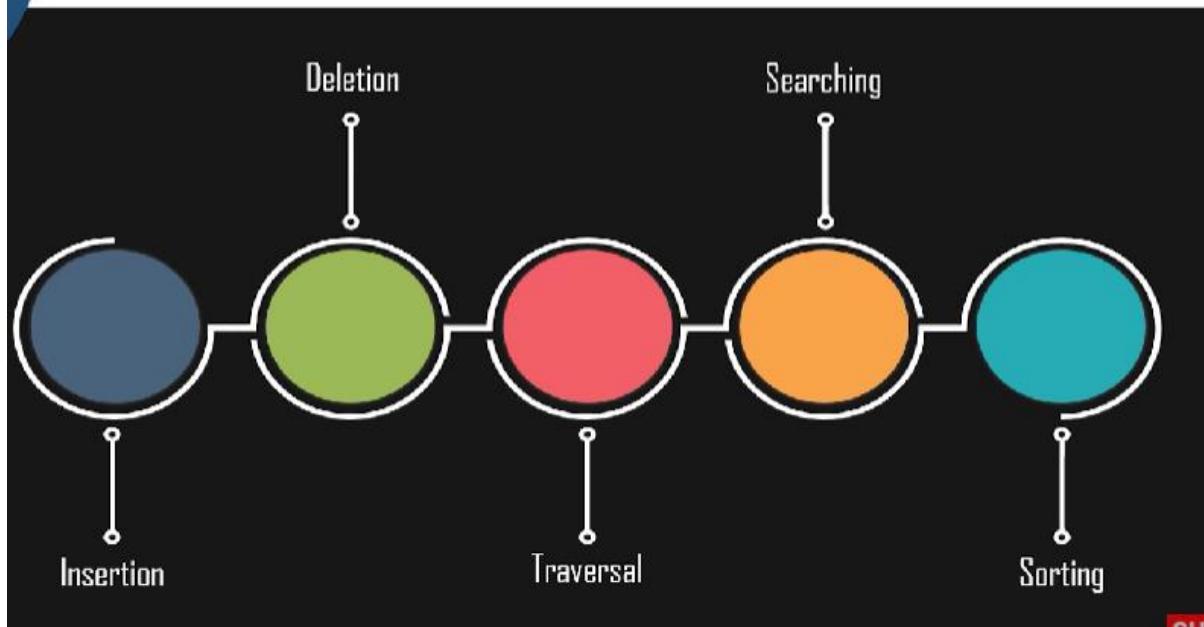
## Explain types of Data Structures



## What are the advantages of Data Structure?



List the operations you can perform on Data Structures



## Differentiate NULL and VOID

Null	Void
It is a Value	It is a data type identifier
Indicates empty value for a variable	Indicates pointer with no initial size
Means never existed	Means existed but not in effect

## How does signed and unsigned numbers affect memory?

Signed
Here, the first bit is used to indicate if the number is positive or negative
Unsigned
Here, you have all bits available for that number
Example: Unsigned 8-bit number has a range 0-255, while the 8-bit signed number has a range -128 to +127.

## Difference between static and dynamic memory allocation

Static Allocation	Dynamic Allocation
Performed at Compile time	Performed at Run time
Assigned to Stack	Assigned to Heap
Size must be known at compile time	Size may be unknown at compile time
FIFO assignment	No particular assignment
Faster execution	Slower execution

SUBSCRIBE

## Explain the role of malloc(), calloc(), realloc() and free()

<b>Malloc()</b> <p>Used for dynamic memory allocation.          Syntax : <code>int *p = (int *)malloc(sizeof(int))</code></p>	<b>Calloc()</b> <p>Used for contiguous dynamic memory allocation.          Syntax : <code>p = (castType*)calloc(n, size);</code></p>
<b>Realloc()</b> <p>Used to resize allocated memory without losing old data.          Syntax : <code>void *realloc(void *p, size_t newsize);</code></p>	<b>Free()</b> <p>Use to free the memory block that had been allocated dynamically.          Syntax : <code>free(p);</code></p>

## Differentiate between Storage and File Structure

Storage Structure	File Structure
Data Structure representation in Computer memory	Storage Structure representation in Auxiliary memory
Example: Memory allocated or a variable or constant	Example: Saving a file in a hard disk

## Differentiate between Recursion vs Iteration

Recursion	Iteration
Function calls itself	A set of instructions repeatedly executed
Terminates when base case is recognized	Terminates when loop condition fails
Used when time complexity is not an issue	Used when time complexity needs to be balanced against an expanded code size
Smaller code size	Large code size

Do all declaration statements lead to fixed memory reservation?

- Most declarations do, with the exemption of pointers.
- Pointer declaration does not allocate memory for data, but for the address of the pointer variable.
- Actual memory allocation for the data comes during run-time.

## How to create an Array?

Arrays can be declared like how a variable is declared

### Declaration

```
int myArray[ ];  
OR  
int[ ] myArray;
```

### Memory allocation

```
int myArray[ ];  
OR  
int[ ] myArray;
```

## Can we change the size of an array at run time?

No ,you can not change the size of your array at run time

### Alternative

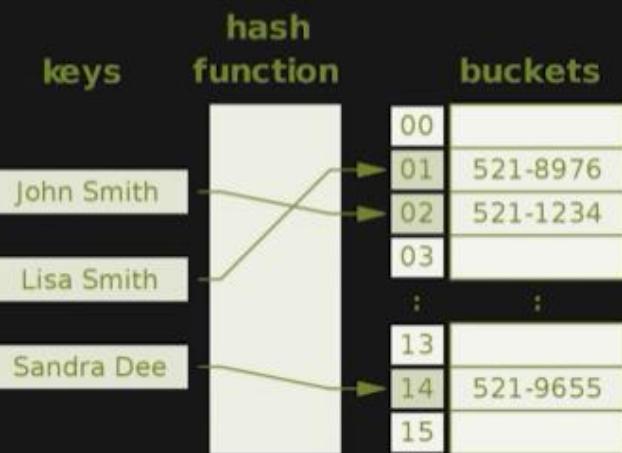
>Create a new array which is larger than the existing one

Copy the elements to the new array

Delete the older existing array

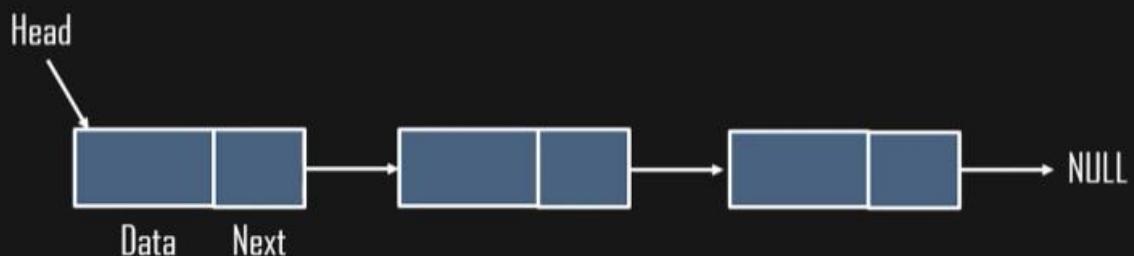
## What do you understand by hashing?

It is a technique of converting a range of key values into a range of indexes of an array.



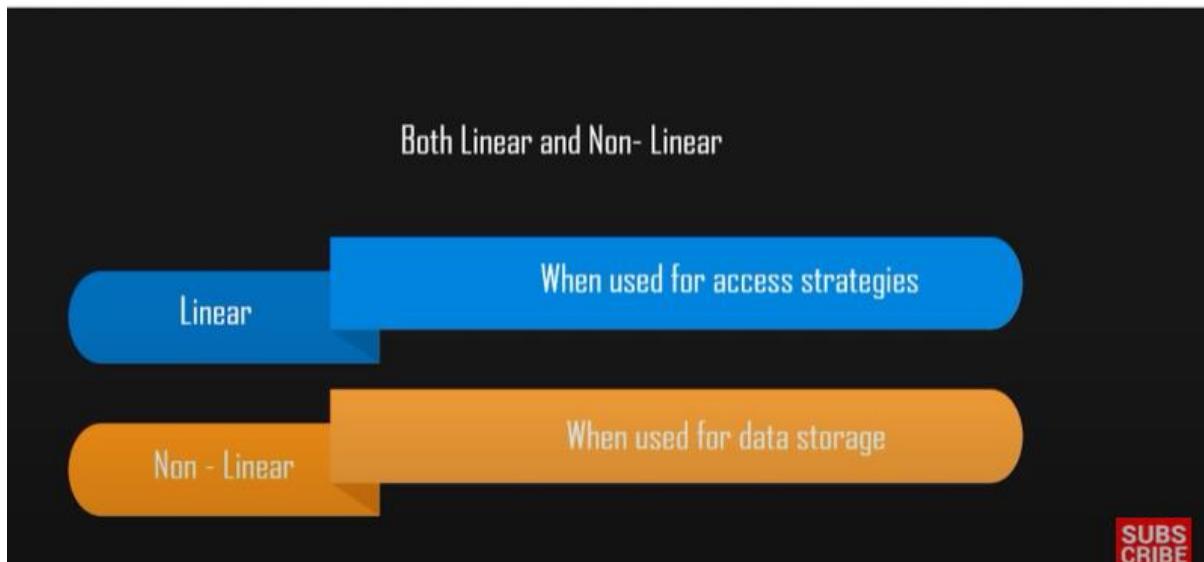
## What is a Linked List?

It is sequence of data structures, which are connected via links. Each link contains a connection to another link.



---

## Are linked lists considered linear or non-linear data structures?



## What is the difference between Array and Linked List?

Array	Linked List
Fixed Size	Dynamic Size
Insertion, deletion of elements is difficult	Insertion, deletion of elements is easy
Random access allowed	Random access not allowed
Better cache locality	Bad cache locality
Chances of memory wastage	No memory wastage

## How do you search for a target key in a Linked List?

Applying Sequential search

### Algorithm:

- Start from the leftmost element of arr[] and one by one compare x with each element of arr[]
- If x matches with an element, return the index
- If x doesn't match with any of elements, return -1

## Explain some applications of Stack

01

Expression Handling

I

Infix to Postfix or Infix to Prefix Conversion

II

Postfix or Prefix Evaluation

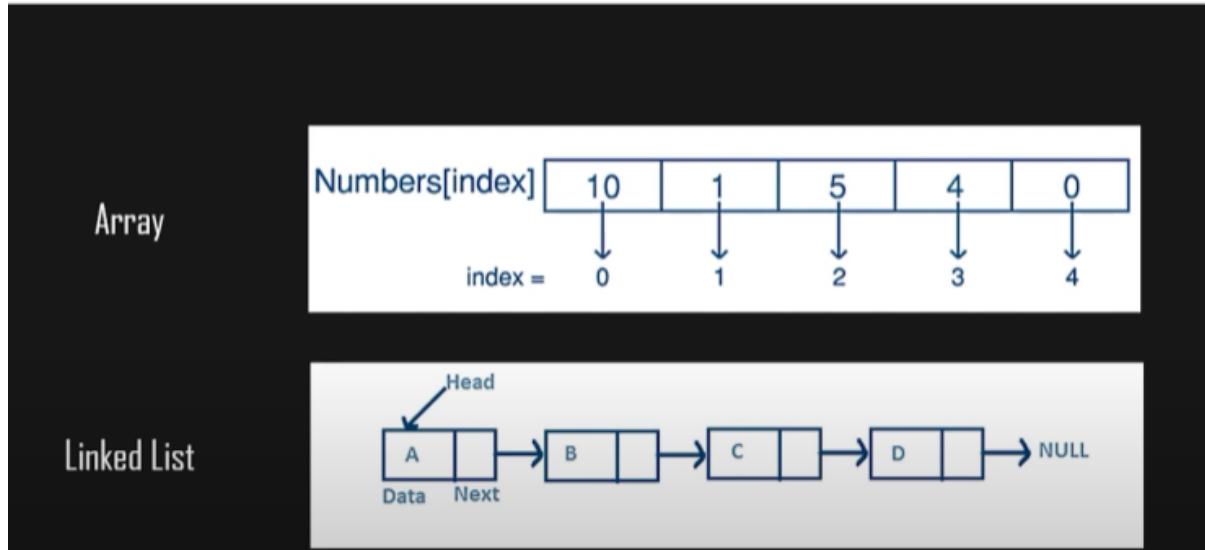
02

Backtracking Procedure

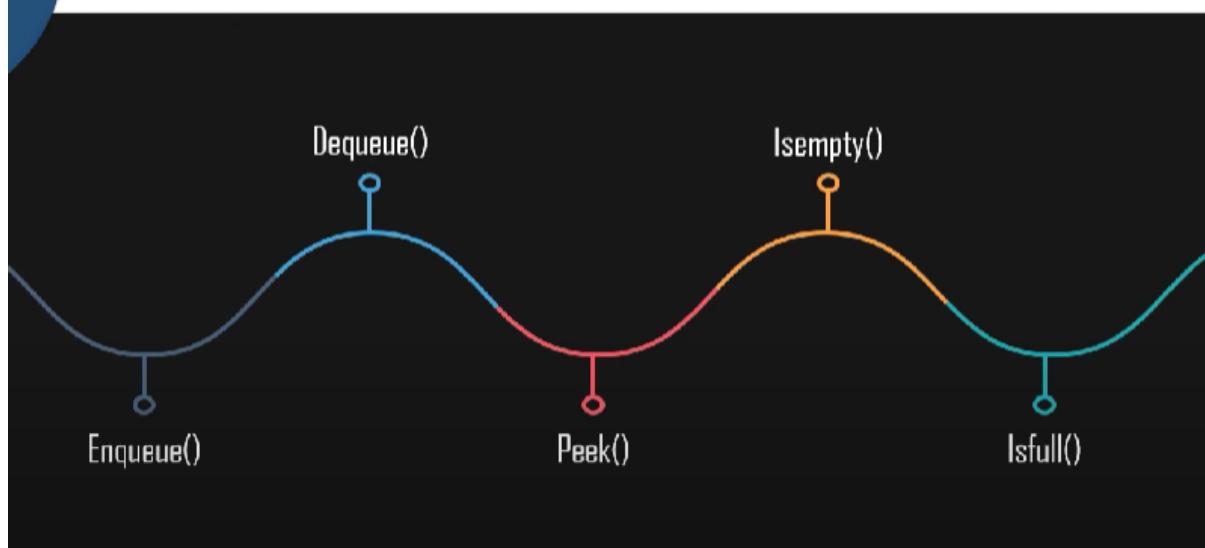
03

Function call and return process

In how many ways we can implement stack?

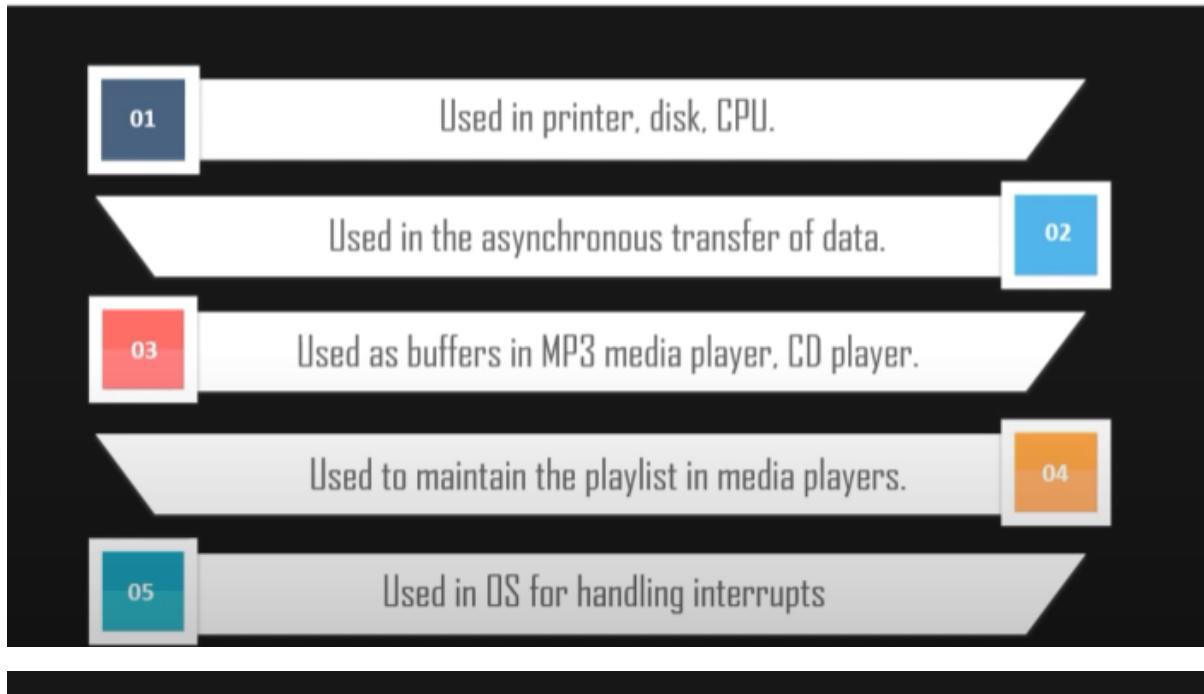


Which operations can be performed on Queue?



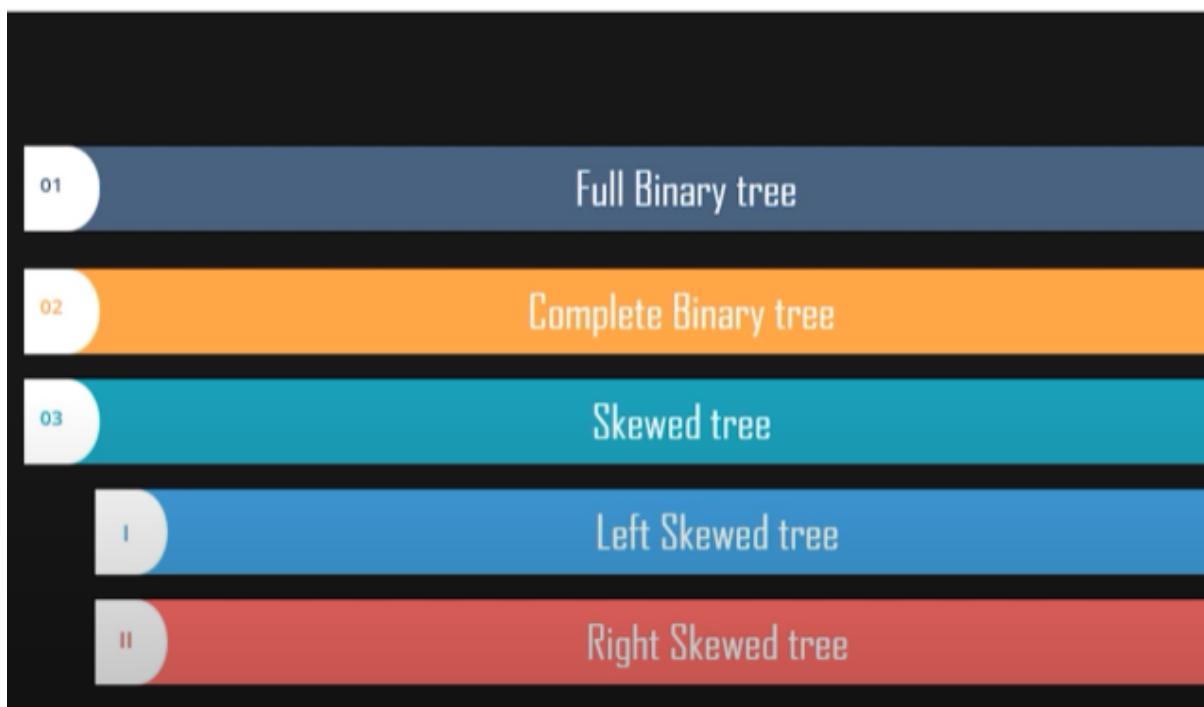
---

## List some applications of queue data structure.



---

## Explain different types of binary tree?

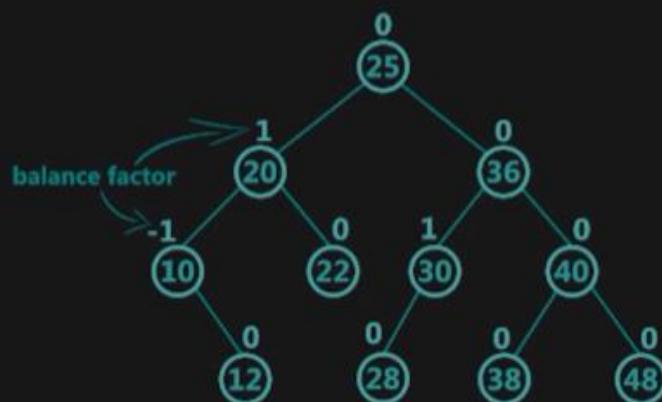


## What are the differences between B tree and B+ tree?

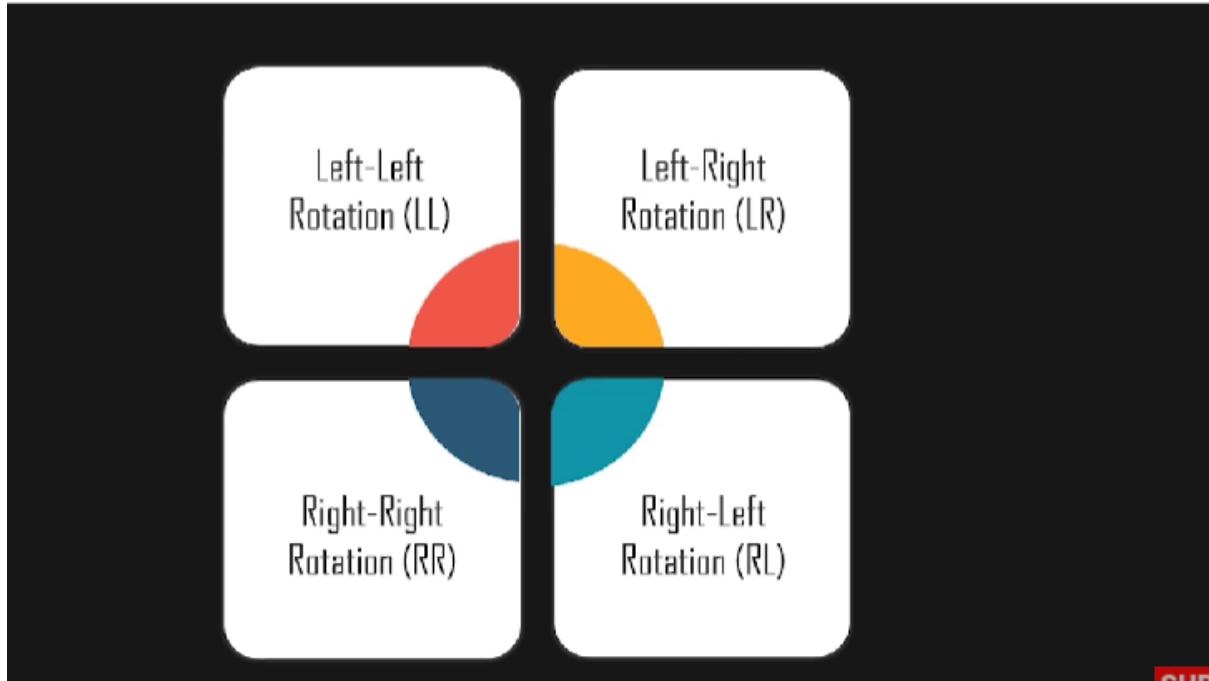
B Tree	B+ Tree
Search keys cannot repeatedly be stored	Redundant search keys can be present
Data can be stored in leaf nodes and internal nodes	Data can only be stored on the leaf nodes
Deletion of internal nodes is complex	Deletion will never be a complexed process
Leaf nodes cannot be linked together	Leaf nodes are linked together to make the search operations more efficient

## What is an AVL tree?

It 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.



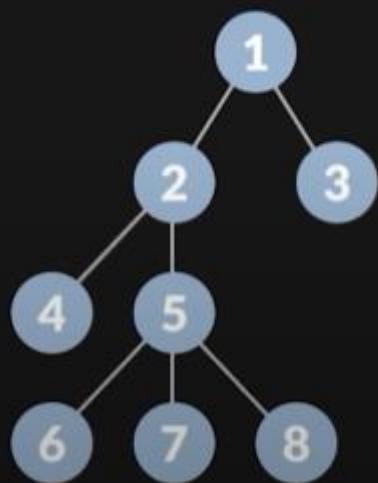
What are the different operations applied on AVL tree?



SUF

What is the degree of a node?

It can be defined as the number of children a node has.



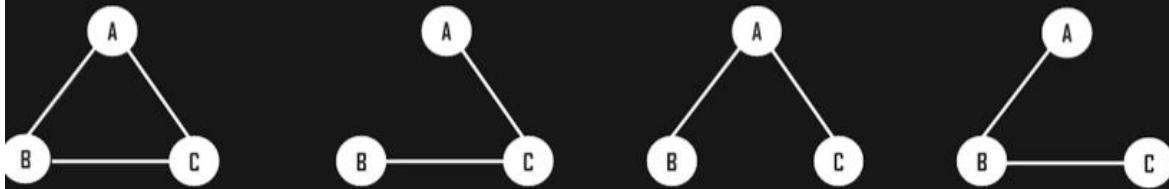
Node 1 degree = 2

Node 2 degree = 2

Node 5 degree = 3

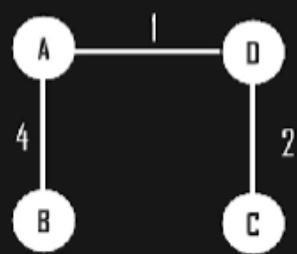
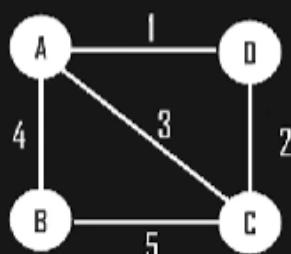
## Explain spanning tree

It is a subset of Graph G, which has all the vertices covered with minimum possible number of edges.



## What is a minimum spanning tree?

Minimum spanning tree is a weighted spanning tree where the cost is minimum among all the spanning trees.



## Differentiate among cycle, path, and circuit

### Path

It is a sequence of adjacent vertices connected by the edges

### Cycle

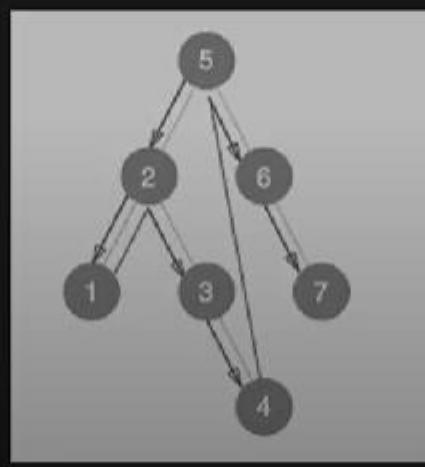
It is a closed path where any vertex in the path can not be visited twice

### Circuit

It is a closed path where any vertex may be repeated

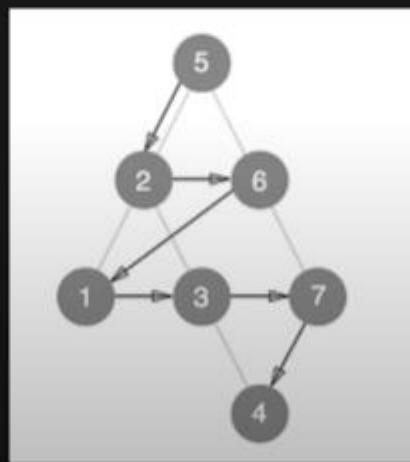
## How does depth-first traversal work?

Here, graph is traversed in a depth ward motion. Stack to remember to get the next vertex.



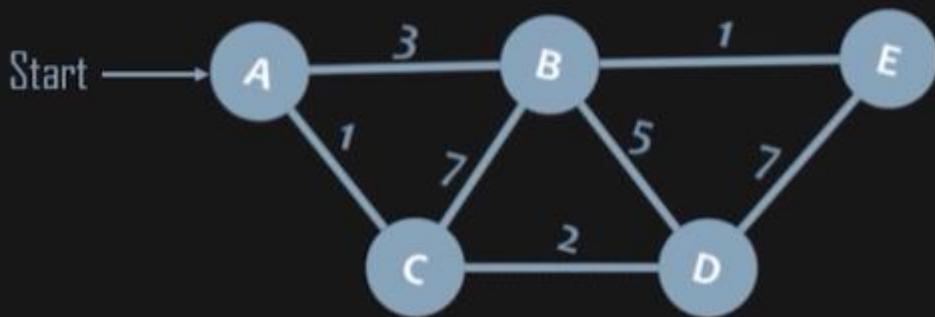
## How does breadth-first traversal work?

Here, graph is traversed in a breadth wards motion and uses a queue to remember to get the next vertex.

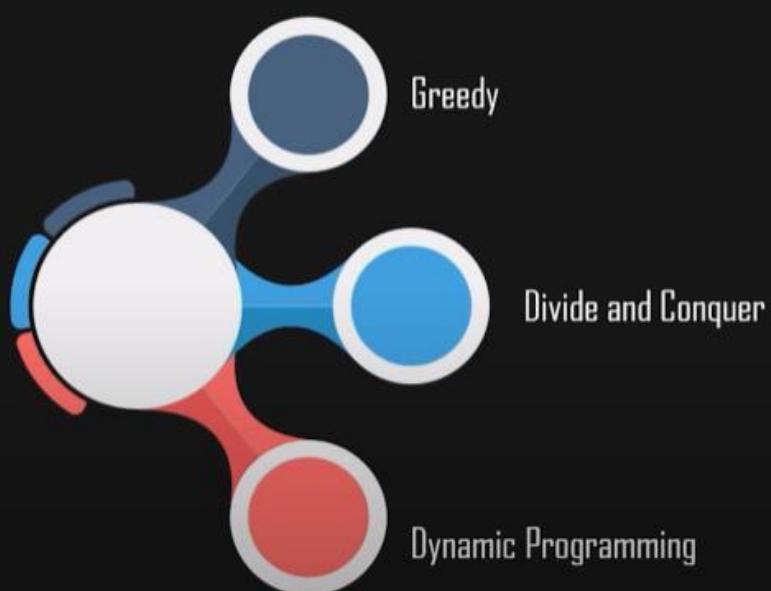


## Explain Dijkstra's algorithm

This algorithm is used to calculate the shortest path between one node of your choice and every other node in a graph



What are different approaches to develop algorithms?



## Explain Greedy approach

It is an approach used to builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit.

### Examples

Dijkstra Algorithm

Prim 's Algorithm

## Explain Divide & Conquer approach

Using this approach, the problem is divided into smaller sub-problems and then each problem is solved independently.

### Examples

Binary Search

Merge Sort

---

## Explain Dynamic Programming

Dynamic programming is used to find the most optimized solution by eliminating the standard recursive calls.

### Examples

Finding Fibonacci series

Knapsack problem

## What is Huffman's algorithm?

It is a lossless data compression algorithm. In this algorithm, a variable-length code is assigned proportional to the character's frequency.

### Examples

String - AAABBC

# 08

## What is an interpolation search technique?

Interpolation search is an improved variant of binary search.

1. Start searching data from the middle of the list.
2. If it is a match, return the index of the item, and exit.
3. If it is not a match, probe position.
4. Divide the list using a probing formula and find the new middle.
5. If data is greater than middle, search in higher sub-list.
6. If data is smaller than middle, search in lower sub-list.

It is technique that sorts the elements by first grouping the individual digits of the same place value. Then, sort the elements.

[121, 432, 564, 23, 1, 45, 788].

1 2 1	0 0 1	0 0 1
0 0 1	1 2 1	0 2 3
4 3 2	0 2 3	0 4 5
0 2 3	4 3 2	1 2 1
5 6 4	0 4 5	4 3 2
0 4 5	5 6 4	5 6 4
7 8 8	7 8 8	7 8 8

sorting the integers according to units, tens and hundreds place.

# What are the disadvantages of Radix sort?

Less flexible

Constant for Radix sort is greater

Consumes more space

## Mathematical puzzle

Tower of Hanoi



It is a mathematical puzzle where we have three rods and n disks. The motto is to move the entire stack to another rod, obeying the following simple rules:

1. One disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another.
3. No disk may be placed on top of a smaller disk.

## Arrays VS Linked Lists



Read – O(1)

Insert – O(n)

Delete – O(n)



Read – O(n)

Insert – O(1)

Delete – O(1)

## HashMaps



## Key-Value Pairs

# HashMaps

45

21

67

41

56

"cats"

"dogs"

"birds"

"fish"

"lions"

## Time Complexity

Read – O(1)

Insert – O(1)

Delete – O(1)

# HashMaps

HashMaps = Hash tables = Dictionaries



# Stacks



**LIFO**

Last in, first out

**Time Complexity**

Push – O(1)

Pop – O(1)

Peek – O(1)

# Queues



**FIFO**

First in, first out

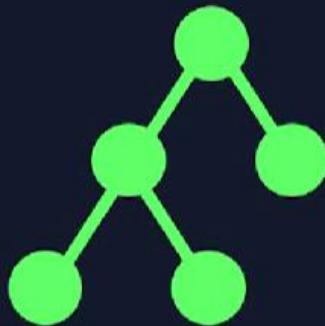
**Time Complexity**

Enqueue – O(1)

Dequeue – O(1)

Front – O(1)

# Binary Search Trees

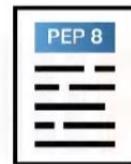


## Time Complexity

Search through every word –  $O(n)$

Binary search –  $O(\log n)$

What is PEP 8 and why is it important?



PEP stands for Python Enhancement Proposal.

A PEP is an official design document providing information to the Python community, or describing a new feature for Python or its processes. PEP 8 is especially important since it documents the style guidelines for Python Code.



Python is an interpreted language. Explain.



An interpreted language is any programming language that executes its statements line by line. This means the source code of a Python program is converted into bytecode that is then executed by the Python virtual machine.

Python is an interpreted language. Explain.



Python code is fast to develop



Python code is not as fast in execution

The code is not directly compiled and executed and an additional layer of the Python virtual machine is responsible for the execution. And little slow as compared to conventional languages like C, C++, etc

What is a dynamically typed language?



Typing refers to type-checking in programming languages. In a strongly-typed language, such as Python, "1" + 2 will result in a type error since these languages don't allow for "type-coercion"

## What is a dynamically typed language?



**Static**

**Data Types are checked before execution.**



**Dynamic**

**Data Types are checked during execution.**

## Is indentation required in Python?



Indentation in Python is compulsory and is part of its syntax. Indentation provides better readability to the code, which is probably why Python has made it compulsory.

## What are Keywords in Python?



Python keywords are special reserved words that have specific meanings and purposes and can't be used for anything but those specific purposes.

## What are Keywords in Python?

False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

As of Python 3.10, there are 35 keywords in Python.

### List all the built-in data types in Python language.

Text Type	:	str
Numeric Types	:	int, float, complex
Sequence Types	:	list, tuple, range
Mapping Type	:	dict
Set Types	:	set, frozenset
Boolean Type	:	bool
Binary Types	:	bytes, bytearray, memoryview
None Type	:	NoneType

These are the built-in Data Types of Python. If you want to know the Data Type of a particular variable, you can use the type() function.

```
a = 8.9
print(type(a))
```

Input



<class 'float'>

Output

These are the built-in Data Types of Python. If you want to know the Data Type of a particular variable, you can use the type() function.

Aspect	Arrays	Lists
<b>Size</b>	Arrays have a fixed size set during creation.	Lists are dynamic and can change in size during runtime.
<b>Data Types</b>	All elements in an array must be of the same data type.	Lists can accommodate elements of different data types.
<b>Memory Allocation</b>	Memory for the entire array is allocated at once during initialization.	Lists dynamically allocate memory as elements are added or removed.
<b>Access Time</b>	Arrays provide constant-time access to elements through indexing.	Lists may have slightly variable access time due to dynamic resizing.
<b>Flexibility</b>	Arrays are less flexible as their size cannot be easily changed.	Lists are more flexible, allowing easy addition or removal of elements.
<b>Memory Efficiency</b>	May lead to memory wastage if size is larger than necessary.	More memory-efficient due to dynamic allocation.

import numpy as np

```
my_arr=np.array(["hello", 1, 2, 3])
print(my_arr)
print(type(my_arr))
Input
```

➤ Arrays need to be declared

➤ Arrays can store data very compactly

➤ Arrays are great for numerical operations

['hello' '1' '2' '3']
<class 'numpy.ndarray'>

Output

On the other hand, NumPy arrays support different data types. To create a NumPy array, you only need to specify the items.

What is the dictionary data type in Python?

Dictionary



Python dictionary is an unordered collection of items. Each item of a dictionary has a key/value pair. Dictionary elements are accessed via keys. You cannot access them using indices like in other data types like lists and tuples.

Explain the concept of indexing in Python. Talk about negative indexing.



It is a special type of object in Python that you can iterate over. It can be easily achieved using 'for' loops. Objects like lists, tuples, sets, dictionaries, strings, etc. are called iterables.

Explain the concept of indexing in Python. Talk about negative indexing.

Indexing in Python is a way to access individual items within an iterable by their position. In other words, you can directly access your elements of choice within an iterable and do various operations depending on your needs.

### Explain the concept of slicing

**Input**

```
example = [1,2,3,4,5]      # create a list
print (example[0:5])       # Whole list, prints [1,2,3,4,5]
print (example[1:5])       # prints [2,3,4,5]
print (example[4:5])       # prints [5]
```

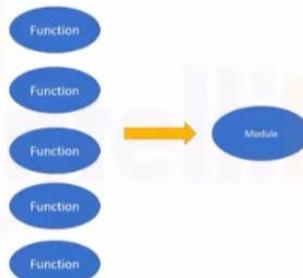
**Slicing a list gives us another list, instead of a single element. A slice specifies a start index and an end index and creates and returns a new list based on the values of the indices. The indices are separated by a colon ':'.**

LIST	TUPLE
<a href="#">Lists</a> are <a href="#">mutable</a>	<a href="#">Tuples</a> are <a href="#">immutable</a>
The implication of iterations is Time-consuming	The implication of iterations is comparatively Faster
The list is better for performing operations, such as insertion and deletion.	A Tuple data type is appropriate for accessing the elements
Lists consume more memory	Tuple consumes less memory as compared to the list
Lists have several built-in methods	Tuple does not have many built-in methods.
Unexpected changes and errors are more likely to occur	Because tuples don't change they are far less error-prone.

### What are functions in Python? Explain with an example.

**We can create a Python function using the def keyword. You can choose any identifier of your choice for the function name. Then comes the parameters which are enclosed within parenthesis. Docstring stands for documentation.**

## Explain Modules and Packages in Python



**By looking at this example we can say that a group of functions which is present inside a python file will make up a module.**

**A module is essentially a python file with a .py extension. It can contain any number of functions, classes and variables.**

**Now, if we want to add complex functionalities other than simple binary operations, we can create another file and import it as another module. For this reason we consolidate all the related modules inside a directory and call it a package.**



**And if you want to work on a very complex project which deals with databases, web based functionalities, linear algebraic operations, trigonometric functions, etc. you will have to consolidate several packages and put it under a single directory and this is what we call a library.**

### Demonstrate any 3 ways to reverse a string in Python

- Using the for loop
- Using the inbuild reversed() function
- Using extended slice syntax

There is no reverse() function for the string library in Python. Therefore we have to know these methods to reverse a string. The following are the 3 ways:

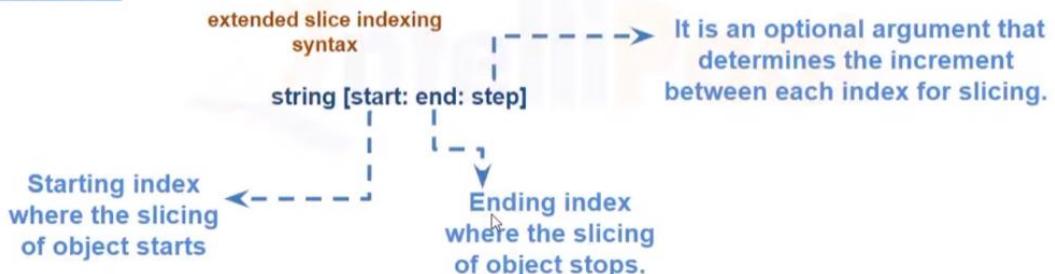
#### Inbuilt reversed()

```
s = "intellipaat"
result = "".join(reversed(s))
print(result)
```

#### Input

In reversed() function which basically creates a reversed object (which is an iterator) from the original string. On wrapping the join() function around it will prepend each character to the empty string on which the dot operator is used

#### Extended slice



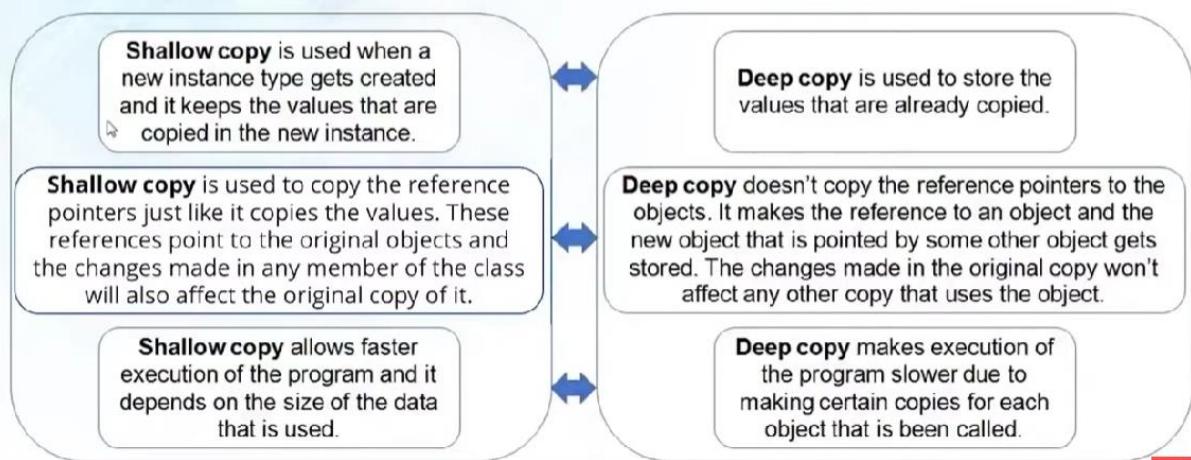
#### Extended slice

```
s = "intellipaat"
print(s[::-1])
```

#### Input

The argument for the step is -1. This will imply start from the last character (-1 is the negative index for the last character in a string) and take one step backwards until the first character of the string.

Q1. What is the difference between deep and shallow copy?



```

>>> l3 = copy.copy(l1)
>>> l3
[1, 2, 3, 4, 10]
>>> l3.append(11)
>>> l1
[1, 2, 3, 4, 10]
>>> l4 = copy.deepcopy(l2)
  
```

Q3. How Multithreading is achieved in Python?

- Python has a multi-threading package but if you want to multi-thread to speed your code up.
- Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only one of your 'threads' can execute at any one time. A thread acquires the GIL, does a little work, then passes the GIL onto the next thread.
- This happens very quickly so to the human eye it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core.
- All this GIL passing adds overhead to execution.

Q5. What is monkey patching in Python?

In Python, the term monkey patch only refers to dynamic modifications of a class or module at runtime. Consider the below example:

**Q9. Explain split(), sub(), subn() methods of "re" module in Python.**

To modify the strings, Python's "re" module is providing 3 methods. They are:

- ❑ **split()** – uses a regex pattern to "split" a given string into a list.
- ❑ **sub()** – finds all substrings where the regex pattern matches and then replace them with a different string
- ❑ **subn()** – it is similar to sub() and also returns the new string along with the no. of replacements.

**Q11. Explain Inheritance in Python with example?**

Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class. They are different types of inheritance supported by Python:

- ❑ Single Inheritance – where a derived class acquires the members of a single super class.
- ❑ Multi-level inheritance – a derived class d1 is inherited from base class base1, and d2 is inherited from base2.
- ❑ Hierarchical inheritance – from one base class you can inherit any number of child classes
- ❑ Multiple inheritance – a derived class is inherited from more than one base class.

```
class ParentClass(object):
    v1 = "from ParentClass - v1"
    v2 = "from ParentClass - v2"

class ChildClass(ParentClass):
    pass

print(ChildClass.v1)
print(ChildClass.v2)
```

### 3. What is the difference between Error and Exception Handling in Python?

Error handling is an attempt to plan for possible errors, it is to let script continue with other code, even if there is an error.

```
In [143]: number1 = 10
In [144]: number2 = input('Please provide a number:')
Please provide a number:20
In [145]: add(number1, number2)

-----
TypeError                                 Traceback (most recent call last)
Input In [145], in <cell line: 1>()
----> 1 add(number1, number2)

Input In [141], in add(n1, n2)
      1 def add(n1,n2):
      2     print(n1+n2)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```



No Exception: Code throwing the error



```
In [146]: try:
    #Want To Attempt This Code
    #May have an error
    result = 10 + '10'
except:
    print('Hey it looks like you are not adding correctly!')

Hey it looks like you are not adding correctly!
```

With Exception: Try and Except Block

## 5. What does List Comprehensions mean in Python?

List Comprehensions are a unique way of quickly creating a list with Python.

```
In [123]: mystring = 'hello'

In [125]: mylist = []

for letter in mystring:
    mylist.append(letter)

In [126]: mylist
Out[126]: ['h', 'e', 'l', 'l', 'o']

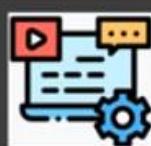
In [127]: mylist1 = [letter for letter in mystring]
mylist1
Out[127]: ['h', 'e', 'l', 'l', 'o']

In [128]: mylist2 =[x for x in 'word']
In [129]: mylist2
Out[129]: ['w', 'o', 'r', 'd']
```



## 15. What is lambda function in Python?

A lambda function is an anonymous function that can take any number of arguments but can have only one expression.



```
In [60]: def square(x):
    return x ** 2
# In regular way we write the squaring function as beside
print(square(10))
100
```

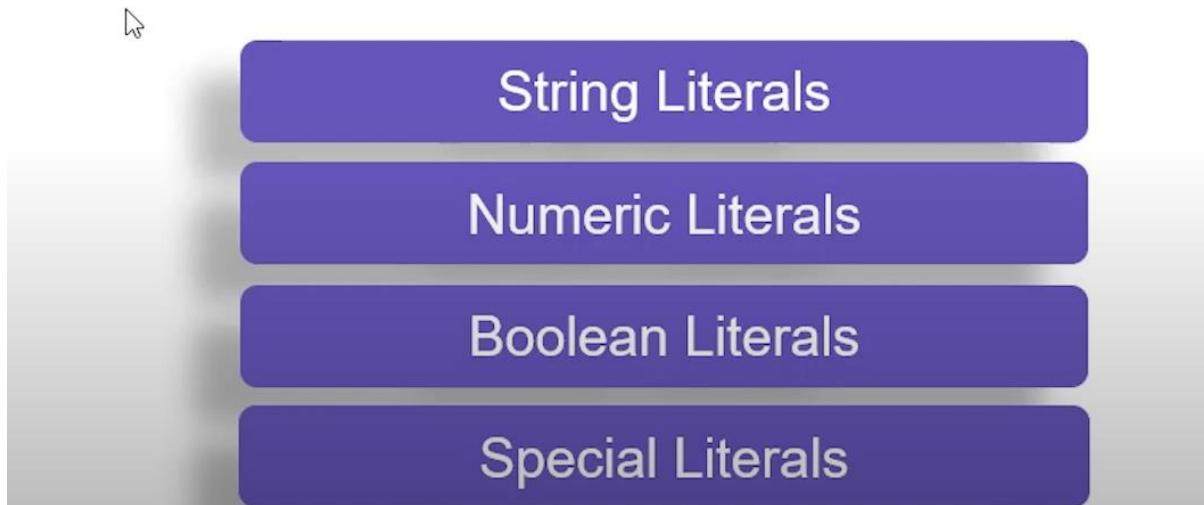
Regular Function

```
In [61]: square = lambda x: x**2
# Using lambda we write the squaring function as beside
print(square(10))
100
```

Anonymous Function

What are literals in Python and explain about the different literals

Literals are the constants used in Python



In Python, `self` is used to refer to the instance of the class in which a method or attribute is being called. It is a convention and not a keyword, but it is a critical part of defining and working with object-oriented programming in Python.

#### Why `self` is Used?

##### 1. Access Instance Attributes and Methods

- **`self` allows access to the attributes and methods of the specific instance of the class.**
- **When you use `self.attribute`, it means you're referring to the attribute belonging to that particular object.**

```
class MyClass:
```

```
    def __init__(self, name):
```

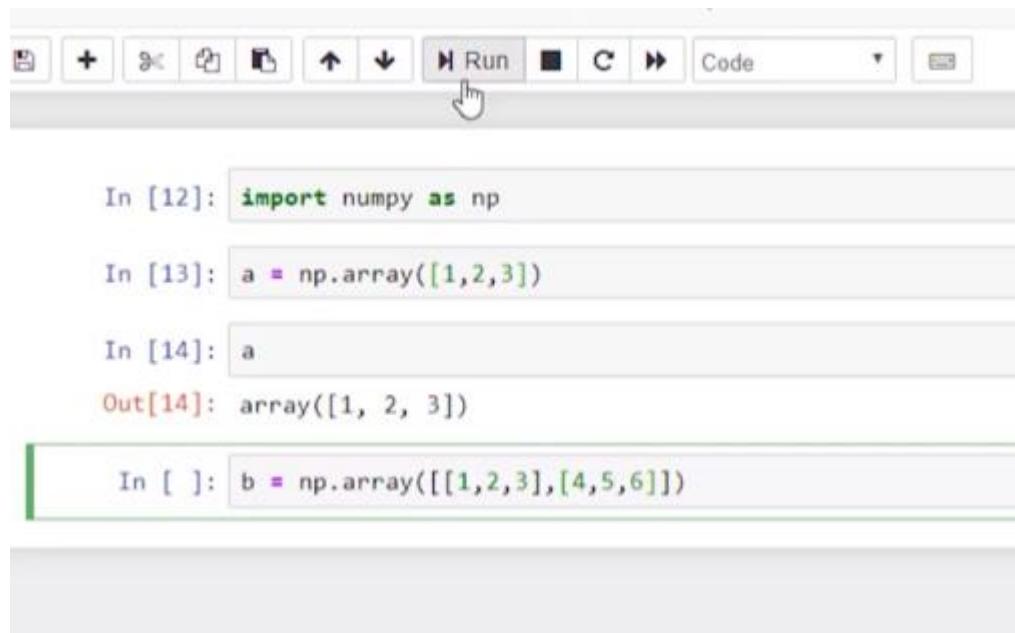
```
        self.name = name # Instance attribute
```

```
    def greet(self):
```

```
        print(f"Hello, {self.name}!") # Access instance attribute
```

```
obj = MyClass("Alice")
```

```
obj.greet() # Output: Hello, Alice!
```



The screenshot shows a Jupyter Notebook interface. At the top is a toolbar with various icons: file, new, cell, cell type, cell selection, run, stop, continue, and code. A mouse cursor is hovering over the 'Run' button. Below the toolbar are four code cells:

- In [12]: `import numpy as np`
- In [13]: `a = np.array([1,2,3])`
- In [14]: `a`
- In [ ]: `b = np.array([[1,2,3],[4,5,6]])`

The cell In [14] has its output displayed below it: `array([1, 2, 3])`. The cell In [ ] is currently active, indicated by a vertical bar on its left.

### **Data Manipulation and Analysis**

1. NumPy - Provides support for numerical computing and working with arrays.
2. Pandas - Enables data manipulation and analysis using DataFrames.
3. SciPy - Offers advanced mathematical, scientific, and engineering functions.

### **Visualization**

4. Matplotlib - Creates static, interactive, and animated visualizations.
5. Seaborn - Builds on Matplotlib to simplify the creation of attractive statistical plots.
6. Plotly - Generates interactive plots, dashboards, and web-based visualizations.

### **Machine Learning and AI**

7. Scikit-learn - Provides simple and efficient tools for machine learning and data mining.
8. TensorFlow - Enables machine learning and deep learning using neural networks.
9. PyTorch - A deep learning framework known for its flexibility and dynamic computation.
10. Keras - A high-level neural networks API running on top of TensorFlow.
11. XGBoost - Implements gradient boosting for efficient and scalable machine learning models.

### **Natural Language Processing**

12. NLTK - Offers tools for text processing and natural language understanding.
13. SpaCy - Provides industrial-strength natural language processing capabilities.
14. TextBlob - Simplifies processing textual data with easy-to-use APIs.

### **Web Development**

15. Flask - A lightweight framework for building web applications.
16. Django - A full-stack framework for rapid web development.
17. FastAPI - A modern framework for building APIs with high performance.

### **Web Scraping**

18. BeautifulSoup - Parses HTML and XML for easy web scraping.
19. Scrapy - A framework for web scraping and crawling.

### **Data Visualization and Dashboards**

20. Dash - Simplifies building interactive web applications with Python.
21. Altair - Offers declarative statistical visualization.

### **Big Data and Parallel Processing**

22. PySpark - Integrates Python with Apache Spark for big data processing.

**23. Dask - Enables parallel computing and scalable data manipulation.**

#### **Image and Video Processing**

**24. OpenCV - Processes images and videos for computer vision tasks.**

**25. Pillow (PIL) - Provides tools for opening, editing, and saving images.**

#### **Scientific Computing**

**26. SymPy - Performs symbolic mathematics in Python.**

**27. Statsmodels - Conducts statistical modeling and hypothesis testing.**

#### **Network and API Handling**

**28. Requests - Simplifies making HTTP requests in Python.**

**29. Socket - Enables low-level networking and communication.**

#### **System and File Handling**

**30. OS - Provides functions for interacting with the operating system.**

**31. Shutil - Offers high-level operations for file and directory management.**

#### **Testing**

**32. PyTest - Simplifies writing and running test cases for Python code.**

**33. Unittest - Provides a testing framework built into Python.**

#### **Others**

**34. Logging - Built-in library for tracking events in applications.**

**35. ConfigParser - Handles configuration files for applications.**

**36. argparse - Simplifies command-line argument parsing.**

**37. Schedule - Enables job scheduling in Python scripts.**

An **API** (Application Programming Interface) is a set of rules and protocols that allows different software applications to communicate with each other. It defines how requests and responses should be structured, enabling developers to interact with external systems, services, or libraries without needing to know their internal workings.

#### **Types of APIs:**

1. **Web APIs:** Allow interaction over the internet using protocols like HTTP (e.g., REST, SOAP).
2. **Library APIs:** Offer access to functions in a programming library.
3. **Operating System APIs:** Facilitate interactions with system resources.
4. **Database APIs:** Enable communication with databases for querying and updates

Aspect	Object-Oriented Programming (OOP)	Procedural Programming
Approach	Focuses on objects and their interactions.	Focuses on functions and procedures.
Structure	Organized into classes and objects.	Organized into a sequence of procedures or functions.
Encapsulation	Data and methods are encapsulated within objects.	Data and functions are separate.
Reusability	Promotes reusability through inheritance and polymorphism.	Reusability is achieved through function calls but limited.
Data Accessibility	Data is protected using access modifiers (private, public, etc.).	Data is global or local to a function but less secure.
Flexibility	Easier to scale and modify due to modular design.	Becomes complex and harder to manage as the project grows.
Examples of Languages	Python, Java, C++, C#, Ruby.	C, Pascal, Fortran, Basic.
Key Concepts	Involves classes, objects, inheritance, polymorphism, abstraction.	Uses functions, procedures, and modules.
Real-World Modeling	Better for modeling real-world problems.	Less intuitive for real-world scenarios.

The **Software Development Life Cycle (SDLC)** is a systematic process used for planning, creating, testing, and deploying information systems. It ensures high-quality software is delivered efficiently.

### Waterfall Model

- **Description:** A linear and sequential approach where each phase depends on the completion of the previous one.
  - **Phases:**
    1. Requirements
    2. Design
    3. Implementation
    4. Testing
    5. Deployment
    6. Maintenance
  - **Use Case:** Suitable for projects with well-defined and unchanging requirements.
- 

### 2. Agile Model

- **Description:** Iterative and incremental development where small, usable portions of the software are delivered in cycles called sprints.
  - **Principles:**
    - Collaboration
    - Flexibility
    - Frequent delivery
  - **Use Case:** Best for projects requiring adaptability and rapid changes.
- 

### 3. Iterative Model

- **Description:** Starts with basic functionality and adds improvements in iterations until the full system is developed.
  - **Phases:**
    - Initial planning
    - Iterative cycles of design, development, and testing
  - **Use Case:** When requirements are not fully clear at the beginning.
-

#### 4. V-Model (Verification and Validation)

- **Description:** An extension of the Waterfall model where every development phase is matched with a corresponding testing phase.
  - **Structure:**
    - Verification (e.g., Requirement Analysis → Test Planning)
    - Validation (e.g., System Design → System Testing)
  - **Use Case:** Projects demanding high reliability (e.g., medical devices).
- 

#### 5. Spiral Model

- **Description:** Combines iterative development with risk analysis at each cycle.
- **Phases:**
  - Planning
  - Risk analysis
  - Engineering
  - Evaluation
- **Use Case:** Large, complex projects with high risk.

## ABOUT TCS

Jamsheji tata

Cotton trading

Own manufacturing factory of cotton cloth (empress mill)

Mysore silk

Taj

Tata Steel - adhura chhuta

Beta - dorabji tata

Mysore k maharaja ki madad se

India institute of science

First super comp. - param

First aircraft - Hansa

By this institute

## HYDROELECTRIC PLANT

(Tata Power)

Steel k liye ipo 23 cr.

Tisco (tata steel)

Jamshedpur , tata nagar railway station

Jrd tata aaye

Tata Aviation services - airline passenger and cargo services

Soda ash and caustic soda

Aayi (tata salt) - gujrat (tata chemicals)

Air india - Indian hospitality

( Gov. Ne leliya)

Phir vo cosmetics m aayi Lakme - Laxmi

Telco(tata motors) - steam engines and truck manufacturing

Tcs

Ratan tata

Architecture ki padhai ki

Tata Steel join kri

(Pune) 100% Indian car -TATA INDICA

TCS - Administrative services to software services

Nayi car - tata nano (failure)

(Sasti thi )

Tata motors - most revenue

Thankyou started acquiring foreign brands

Jaguar

Land rover

Tetly

8oclock coffee

When was TCS established?

"TCS, or Tata Consultancy Services, was established in 1968."

4. CEO of TCS

"The current CEO of TCS is K Krithivasan (since 2023)."

5. Co-founder of TCS

"TCS was founded by Tata Sons, led by J.R.D. Tata and F.C. Kohli, who is also considered the father of the Indian IT industry."

6. Headquarters of TCS

"The headquarters of TCS is located in Mumbai, Maharashtra, India."

### 7. Which cities are Tier-2 in TCS?

"Some of the Tier-2 cities where TCS operates include Pune, Nagpur, Bhubaneswar, Coimbatore, and Kochi."

#### 1. What is TCS, and what services does it offer?

**Answer:** TCS (Tata Consultancy Services) is a global IT services, consulting, and business solutions company headquartered in Mumbai, India. It is a subsidiary of Tata Group. TCS offers services in:

- **IT Services:** Software development, systems integration, and IT infrastructure management.
  - **Consulting:** Digital transformation, business consulting, process optimization.
  - **Cloud Computing:** Cloud migration, cloud infrastructure management, cloud-native development.
  - **Data & Analytics:** Big data solutions, AI, machine learning, and data visualization.
  - **Enterprise Solutions:** ERP, CRM (SAP, Oracle, Dynamics).
  - **Cybersecurity:** Information security, risk management, compliance services.
  - **Engineering Services:** Product lifecycle management, embedded systems, IoT solutions.
- 

#### 2. What are TCS's core values?

**Answer:** TCS's core values are:

- **Integrity:** Doing what is right in all situations.
  - **Excellence:** Striving for the highest standards in everything.
  - **Unity:** Working together as a team and respecting diverse perspectives.
  - **Responsibility:** Taking ownership of actions and outcomes.
  - **Pioneering:** Innovation and leadership in technology and business solutions.
- 

#### 3. What is the TCS Business Model?

**Answer:** TCS follows a **global delivery model**, which focuses on providing IT services through a combination of offshore, onshore, and nearshore locations. The key components are:

- **Consulting and Strategy:** Identifying business solutions and technology strategies for clients.
- **Technology & Engineering Services:** Providing end-to-end solutions using cutting-edge technologies.
- **Business Process Outsourcing (BPO):** Outsourcing non-core functions like customer service, finance, etc.
- **Digital Transformation:** Helping clients transform their businesses using AI, cloud computing, IoT, and more.

#### 4. What are the key achievements of TCS?

**Answer:** Some of the major achievements of TCS include:

- **Market Leadership:** TCS is among the top 10 largest IT services companies globally.
  - **Global Presence:** TCS operates in more than 50 countries.
  - **Innovative Solutions:** TCS has pioneered several innovations in digital, cloud, and AI-driven solutions.
  - **Awards:** TCS has received numerous industry awards, including the “Best Employer” award and recognition for sustainability practices.
- 

#### 5. What is TCS's approach to innovation?

**Answer:** TCS focuses on "**Innovation through Technology**" by investing heavily in research and development (R&D). It has several innovation labs and centers globally that focus on emerging technologies like AI, machine learning, blockchain, and IoT. TCS has also partnered with global academic institutions to drive innovation.

#### What is TCS's commitment to sustainability and CSR?

**Answer:** TCS is committed to **corporate social responsibility (CSR)** through its initiatives focused on:

- **Education:** Supporting underprivileged children through initiatives like TCS iON and TCS Foundation.
  - **Environment:** TCS has set sustainability goals to reduce carbon emissions and increase the use of renewable energy.
  - **Community Engagement:** TCS employees volunteer in various social development activities.
- 

#### 8. What are the recent trends in technology that TCS is focusing on?

**Answer:** TCS is focusing on the following **technological trends**:

- **AI & Machine Learning:** Implementing AI solutions for predictive analytics, automation, and intelligent decision-making.
- **Cloud Computing:** Helping organizations migrate to the cloud and implement hybrid cloud environments.
- **Blockchain:** Using blockchain for secure, transparent transactions, and supply chain management.
- **Automation & RPA (Robotic Process Automation):** Reducing manual efforts in business processes.
- **Cybersecurity:** Enhancing security measures to protect against modern-day threats.

top clients of TCS along with the services provided:

---

### **1. Cognizant Technology Solutions**

- **Industry:** Information Technology & Services
  - **Services Provided:**
    - **Consulting Services:** IT strategy, business process improvement.
    - **Cloud Solutions:** Implementation of cloud platforms like AWS, Azure.
    - **Digital Transformation:** Application modernization, analytics, automation.
    - **Software Engineering:** Custom software solutions, application development, and support.
- 

### **2. Citibank**

- **Industry:** Banking & Financial Services
  - **Services Provided:**
    - **Banking Solutions:** Core banking systems and transaction services.
    - **Digital Banking Services:** Mobile banking solutions, UX/UI improvements.
    - **Cybersecurity Services:** Fraud detection, risk management.
    - **Data Analytics & AI:** Predictive analytics for financial products, customer insights.
- 

### **3. Volkswagen Group**

- **Industry:** Automotive
  - **Services Provided:**
    - **Digital Transformation:** Cloud migration, IoT implementation for connected vehicles.
    - **Enterprise Solutions:** ERP systems, legacy system modernization.
    - **Data Analytics & Machine Learning:** Predictive maintenance, supply chain optimization.
    - **Automation & Robotics:** Robotic Process Automation (RPA) for manufacturing processes.
-

#### 4. General Electric (GE)

- **Industry:** Industrial & Manufacturing
  - **Services Provided:**
    - **IoT Solutions:** Smart manufacturing solutions, IoT-driven insights.
    - **Cloud & Data Analytics:** Big data processing, cloud infrastructure support.
    - **Engineering & IT Consulting:** Product lifecycle management, system integration.
    - **Cybersecurity Services:** Protection for industrial assets and operations.
- 

#### 5. Microsoft

- **Industry:** Technology
  - **Services Provided:**
    - **Cloud Services:** Microsoft Azure migration, cloud-native solutions.
    - **Enterprise Solutions:** ERP and CRM systems (SAP, Dynamics 365).
    - **Digital Experience Solutions:** Design and development of customer-facing apps and platforms.
    - **AI & Data Analytics:** Implementation of AI models and data visualization tools.
- 

#### 6. American Express

- **Industry:** Financial Services
  - **Services Provided:**
    - **Card Services Solutions:** Fraud prevention, card processing systems.
    - **Digital Payments:** Mobile payments, wallet integration, and e-commerce solutions.
    - **Data Security & Compliance:** PCI-DSS compliance, cybersecurity.
    - **Customer Experience:** Loyalty programs, chatbot development, and customer service automation.
- 

#### 7. Shell

- **Industry:** Energy & Petrochemical
- **Services Provided:**
  - **Digital Transformation:** Cloud migration, infrastructure modernization.
  - **Data Analytics:** Energy management and predictive analysis.
  - **IoT Solutions:** Real-time monitoring of machinery and equipment in oil fields.

- **Blockchain Solutions:** Energy trading platforms, secure supply chain tracking.
- 

## 8. Unilever

- **Industry:** Consumer Goods
  - **Services Provided:**
    - **Supply Chain Optimization:** End-to-end supply chain management, logistics.
    - **Enterprise Solutions:** SAP ERP systems, digital transformation of operations.
    - **Marketing Analytics:** Consumer insights, digital marketing, and campaign management.
    - **Sustainability Solutions:** Data-driven solutions for sustainable sourcing and product lifecycle management.
- 

## 9. BMW Group

- **Industry:** Automotive
  - **Services Provided:**
    - **Connected Vehicles Solutions:** IoT, autonomous vehicle technology.
    - **Cloud Solutions:** Cloud infrastructure for real-time data processing.
    - **Data Analytics & AI:** Predictive analytics for vehicle maintenance, customer behavior modeling.
    - **Digital Transformation:** AI-based manufacturing and supply chain optimization.
- 

## 10. Hershey

- **Industry:** Food & Beverage
- **Services Provided:**
  - **SAP Solutions:** ERP implementation and integration across functions.
  - **AI & Data Analytics:** Demand forecasting, customer insights.
  - **Cloud Services:** Migration to cloud platforms for better scalability.
  - **Cybersecurity:** Protecting consumer data and intellectual property.