Technical Subjects commonly asked interview questions

Python
Video Link :

Question	Answer
1. what is the difference between c and python	**C** is a general-purpose, procedural computer programming language. It is statically typed language C Program syntax is harder than Python. C has a limited number of built-in functions. C programs are compiled by the compiler C is generally used for hardware related applications. **Python** is an interpreted, high-level, general-purpose programming language. It is dynamically typed It is easier to write a code in Python as the number of lines is less comparatively. Python has a large library of built-in functions. interpreter is used in Python Python is a General-Purpose programming language.

2. what is loop, what are the loops used in python	A loop is a sequence of instruction s that is continually repeated until a certain condition is reached
	For loop and while loop are used in python
	The for loop in Python is used to iterate over a sequence (list,tuple,string) or other iterable objects. Iterating over a sequence is called traversal
	Eg:
	for i in range(3):
	    print(i)
	The while loop in Python is used to iterate over a block of code as long as the test expression (condition) is true.
	#print numbers from1 to 5
	i = 1
	while i < 6:
	  print(i)
	  i += 1

3. Difference between lists and tuple	**LIST** List is a collection of ordered elements,surrounded by square brackets[]. List is mutable(changeable) List is dynamic. They grow and shrink on demand. The list is better for performing operations, such as insertion and deletion. Lists consume more memory Lists have several built-in methods **TUPLE** Tuple is collection of ordered elements surrounded by

	round brackets(). It is immutable(unchangeable) Tuples are fixed in length, heterogeneous, and can be nested. Tuple data type is appropriate for accessing the elements Tuple consume less memory as compared to the list Tuple does no have must built-in methods.
4. Difference between mutable and immutable	**Mutable** means, wecan change the values of a variable, which we have stored. **mutable :**List,Set,Dictionary **Note**:Set is mutable, but every element is unique and must be immutable Keys in dictionary are unique and they are immutable (strings,numbers,tuples) **Immutable** means, we cannot change the values of a variable, which we have stored. **Immutable:**Tuple,Strings,Numbers
5. what is list comprehension	**Comprehensions** in Python provide us with a short and concise way to construct new sequences (such as lists, set, dictionary etc.) using sequences which have been already defined. **List comprehension** allows us to create a new list of elements that satisfy a condition from an iterable. An iterable is any Python construct that can be looped over like lists, strings, tuples, sets. In list comprehensions we use square brackets. **s=[x*x for x in range(1,11) if x%2==0]** Result of above comprehension will be: **[4, 16, 36, 64, 100]**

6. what is list slicing	**Slicing** is used to build new lists out of an existing list.(or)Slicing is used to extract part of a list
	We can access a range of elements in a list by using the slicing operator (colon).
	Eg:
	l = [‘a’,’b’,’c’,’d’,’e’,’f’,’g’]
	print(l)
	print(l[1])
	print(l[1:3])
	print(l[1:-1])
	print(l[:3])

7. Use of pip, Full form of pip	pip is a package-management system written in Python used to install and manage software packages. It connects to an online repository of public packages,

	called the Python Package Index. pip stands for "preferred installer program"
8. Null pointer in python	Python Null object is the singleton None. There’s no null value in Python; instead, there’s None. The equivalent of the null keyword in Python is None.
9. Local and global variables	**Global Variable** Variable which is defined in the main body of the program file. Accessible throughout the program file. Accessible to all functions in the program. **Local Variable** Variable which is defined inside a function. Accessible from the point it is defined to the end of the block it is defined in. They are not related in any way to other variables outside the function. **Eg:** x = 20 def fun(): x = 10 print("Local x =",x) fun() print("Global x =",x) **Output of above code is as follows:** Local x = 10 *Global x = 20*
10. Pep 8	Code is read much more often than it is written.PEP 8 exists to improve the readability of Python code.It helps in writing clear and readable code.It helps in understanding the structure of the program
11. Is python scripting language or programming language?	Python is one of the best programming and scripting languages among all. It doesn’t require any compilation and are directly interpreted.The interpreter analyzes the entire program every time and halts the execution if any error is encountered.
12. Self keyword, pass	The **pass statement** is used as a placeholder for future code. When the pass statement is executed, nothing happens, but we avoid getting an error when empty code is not allowed. Empty code is not allowed in loops, function definitions, class definitions, or in if statements **Syntax:** Pass **Using the pass keyword in for loop:** for x in [0, 1, 2]: pass

	Using the pass keyword in a function definition:
	def myfunction:
	        pass
	self represents the instance of the class. By using the “self” keyword we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

13. in , not , and operators in python.	**Python Logical Operators:and,not,or** **And:**returns True if both the statements are True **Or:**returns True if one statement is True **Not:**Reverses the result.returns False if the result is True **#Logical operators** x=int(input("Enter the value of x:")) print(x<5 and x<10) print(x<5 or x<4) print(not(x>10)) Output: Enter the value of x:6 False False True **Python Membership Operators:in,not in** **in** returns True if it finds a variable in the specified sequence and false otherwise. **not in** Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. **#Membership operators** x=[1,2,3,4,5] y=int(input("enter the value of y:")) print(y in x) print(y not in x) **OUTPUT:** enter the value of y:5 True False
14. features of python	1. Easy to Code,read and Expressive 2.Free and Open-Source:Python is freely available. 3. Python is a high-level language. 4. Portable:Write once and execute anywhere 5. Interpreted: source code is executed line by line, and not all at once. 6.Python supports both procedure-oriented and object-oriented programming which is one of the key python features.It also supports multiple inheritance, unlike Java. 7. Python is dynamically-typed. This means that the type for a value is decided at runtime, not in advance. 8. Python is an extensible language, meaning that it can be extended to other languages. 9.Python has large standard libraries in it.

15. Functions used in python and explain lambda function	**Yes,** functions are used in python.It is defined as a block of organized and reusable program code that performs a specific, single, and well-defined task. Understanding programs becomes easier by using the functions. **Anonymous Functions or LAMBDA Functions:**Functions that don’t have any name are known as lambda or anonymous functions. Lambda functions are created using the keyword _lambda_. Lambda functions are one line functions that can be created and used anywhere a function is required. **Syntax:**_lambda arguments-list : expression_ _power = lambda x : x*x*x_ _print(power(3))_ ***#output:27*** · Lambda functions doesn’t have any name. · Lambda functions can take multiple arguments. · Lambda functions can returns only one value, the value of expression. · Lambda function does not have any return statement. · Lambda functions are one line functions.
16. explain call by value and call by reference	Python programming language uses the mechanism of the call-by-object and also call by object reference. If you pass the arguments like strings, tuples to function the passing values will act as the call by value. **Call by value:** A copy of the variable is passed. Change in the copy of the variable,doesn't modify the original value of the variable. Primitive types are passed using call by value **Call by reference:** A variable itself is passed. Change in the copy of the variable, modify the original value of the variable. Objects are implicitly passed using call by reference.
17. explain function and what is recursive function	A function is a block of organized and reusable program code that performs a specific, single, and well-defined task. A function provides an interface for communication in terms of how information is transferred to it and how results are generated. **There are two types of functions**: Built-in-functions User defined functions A recursive functions is a function which calls itself. A recursive functions should contain two major parts: • base condition part: A condition which terminates recursion and returns the result. • recursive part: Code which calls the function itself with reduced data or information.

	Recursion uses divide and conquer strategy to solve problems.
	def fact(n):
	"""Returns the factorial of a number using recursion."""
	if n==0 or n==1:
	return 1
	else:
	return n*fact(n-1)
	
	n=int(input("enter n value"))
	print("factorial=",fact(n))

18. python is case sensitive language?	Python is a case-sensitive language because it distinguishes between identifiers like Variable and variable. In simple words, we can say it cares about uppercase and lowercase.
19. Interpreter vs Compiler	**Compiler** It takes an entire program at a time. Comparatively fast Memory requirement is more due to the creation of object code. Displays all errors after compilation ,all at the same time Error Detection is Difficuilt C,C++,Java,Scala,etc., **Interpreter** It takes a single line of code Slower It requires less memory as it does not create intermediate object code. Displays error of each line one by one Error Detection is Easy Python,Perl,PHP,Ruby,etc.,
20. What is lambda and it's function in programming	**Anonymous Functions or LAMBDA Functions:**Functions that don't have any name are known as lambda or anonymous functions. Lambda functions are created using the keyword *lambda*. Lambda functions are one line functions that can be created and used anywhere a function is required. **Syntax:***lambda arguments-list : expression* *power = lambda x : x*x*x* *print(power(3))* ***#output:27*** · Lambda functions doesn't have any name. · Lambda functions can take multiple arguments. · Lambda functions can returns only one value, the value of expression. · Lambda function does not have any return statement. · Lambda functions are one line functions

21. what is modular programming	**Modular programming** is a way of organizing programs as they become more complicated. we can create a Python module, a source file that contains Python source code to do something useful, and then import this module into our program so that you can use it. Creation of Module: mymodule.py def greeting(name): print("Hello, " + name) Use of Module: import mymodule mymodule.greeting("CRT") Output:Hello,CRT

22. What are Dictionaries. Give examples	**Dictionaries:**
	Python dictionary is an ordered collection of elements and it is mutable.
	We can store values as a pair of keys and values
	Represented by dict() or {}
	Keys in dictionary are unique and they are immutable(strings,numbers,tuples)
	Creating a dictionary
	d={}
	print(d)
	d1={1:'A',2:'B',3:'C'}
	print(d1)
	d2={'a':1,'b':2,'c':3}
	print(d2)
	d3={1:1.1,2:'a',3:[1,2,3],4:(1,2,3)} #mixed dictoinary
	print(d3)

23. Dir, help functions in python	**help()** is a very useful built-in function that can be used to return the Python documentation of a particular object, method, attributes, etc. example my_list = [] help(my_list.append) **output** Help on built-in function append: append(...) method of builtins.list instance L.append(object) -> None -- append object to end **dir()** shows a list of attributes for the object passed in as argument , without an argument it returns the list of names in the current local namespace (similar to locals().keys()) . example 1. my_list = [] 2. print(dir(my_list)) **output** 1. ['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__' 2. , '__doc__', '__eq__', '__format__', '__ge__',

	'__getattribute__', '__getitem__' 3. , '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__iter__', '__le__' 4. , '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_e 5. x__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__s 6. izeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'ex 7. tend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
24. Substring method	Python has no substring methods like substring() or substr(). Instead, we use slice syntax to get parts of existing strings.

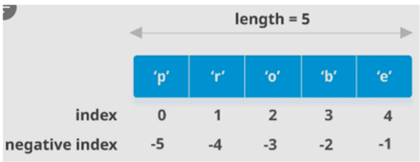
25. Constructor	class Box:
	        def __init__(self, l, b):
	                self.length = l
	                self.breadth = b
	
	        def printdetails(self):
	                print("Length of box is:", self.length)
	                print("Breadth of box is:", self.breadth)
	        def display(self):
	                self.printdetails()
	b = Box(10,20)
	b. display()
	OUTPUT:
	Length of box is: 10
	Breadth of box is: 20

26. Can the constructor be inherited?	In Python, every class inherits from a built-in basic class called ‘object’. The constructor i.e. the ‘__init__’ function of a class is invoked when we create an object variable or an instance of the class. The variables defined within __init__ () are called as the instance variables or objects.
27. difference between array and list	**Arrays:** An array is a data structure that holds fix number of elements and these elements should be of the same data type. There are two important parts of the array: one is an **Element:** Each item stored in the array is called an element and second is an **Index:** Every element in the array has its own numerical value to identify the element. **Lists:** The list is written as the list of commas separated values inside the square bracket. The elements in the list are not compulsory of the same datatype. It can be represented with positive index or negative

	index.
28. Memory mapping in python	Memory mapping is an alternative approach to file I/O that's available to Python programs through the mmap module. Memory mapping uses lower-level operating system APIs to store file contents directly in physical memory.
29. Dynamic memory allocation in python	Dynamic memory allocation is mostly a non-issue in Python. Everything is an object, and the reference counting system and garbage collector automatically return memory to the system when it is no longer being used.

30. python program- class implementation with constructor	class Box:
	        def __init__(self, l, b):
	                self.length = l
	                self.breadth = b
	        def printdetails(self):
	                print("Length of box is:", self.length)
	                print("Breadth of box is:", self.breadth)
	        def display(self):
	                self.printdetails()
	b = Box(10,20)
	b. display()
	OUTPUT:
	Length of box is: 10
	Breadth of box is: 20

31. Built in libraries in python	The most important Python libraries are:Numpy. Pandas. Matplotlib. Scikit-Learn. Scipy.
32. How will you remove duplicates from an array	To remove the duplicates from a list, you can make use of the built-in function set(). The specialty of set() method is that it returns distinct elements. We have a list : [1,1,2,3,2,2,4,5,6,2,1]. The list has many duplicates which we need to remove and get back only the distinct elements. The list is given to the set() built-in function. Later the final list is displayed using the list() built-in function, as shown in the example below. my_list = [1,1,2,3,2,2,4,5,6,2,1] my_final_list = set(my_list) print(list(my_final_list)) **Output:** [1, 2, 3, 4, 5, 6]
33. Negative indexing in python	Negative indices allow you to refer to an item's position relative to the end of the list. This means that the index value of -1 gives the last element, and -2 gives the second last element of list. The negative indexing starts from where the list ends.

	 length = 5 'p' 'r' 'o' 'b' 'e' index 0 1 2 3 4 negative index -5 -4 -3 -2 -1
34. Data types in python	Text Type: str Numeric Types: int, float, complex Sequence Types: list, tuple, range Mapping Type: dict Set Types: set, frozenset Boolean Type: bool

35. Asked a program to write multiplication table using python	`# Multiplication table (from 1 to 10) in Python`
	
	`num = 12`
	
	`# To take input from the user`
	`# num = int(input("Display multiplication table of? "))`
	
	`# Iterate 10 times from i = 1 to 10`
	`for i in range(1, 11):`
	`   print(num, 'x', i, '=', num*i)`

C/C++
Video Link: :

Question	Answer
1. String reverse logic	● Input the string from the user ● Find the length of the string. The actual length of the string is one less than the number of characters in the string. Let actual length be j. ● Repeat the below steps from i = 0 to the entire length of the string. ● rev[i] = str[j] ● Print the reversed string.

2. write a program to check given string is palindrome?	**Using Predefined String Functions:**
	
	#include <stdio.h>
	#include <string.h>
	int main()
	{
	  char a[100], b[100];

```
  printf("Enter a string to check if it's a palindrome\n");
  gets(a);
  strcpy(b, a);
  strrev(b);
  if (strcmp(a, b) == 0)
    printf("The string is a palindrome.\n");
  else
    printf("The string isn't a palindrome.\n");

  return 0;
}
```

Without Using Functions

```
#include <stdio.h>
#include <string.h>
int main() {
            char text[100];
            int begin, middle, end, length = 0;
            gets(text);
            while ( text[length] != '\0' )
                  length++;
            end = length - 1;
            middle = length/2;
            for ( begin = 0 ; begin < middle ; begin++ ) {
                        if ( text[begin] != text[end] ) {
                                    printf("Not a palindrome.\n");
                                    break;
                        }
                        end--;
            }
            if( begin == middle )
                  printf("Palindrome.\n");
            return 0;
}
```

3. What is structure

Structure in c is a user-defined data type that enables us to store the collection of different data types.

Each element of a structure is called a member.

The ,struct keyword is used to define the structure. Let's see the syntax to define the structure in c.

struct structure_name

	{ data_type member1;
	
	    data_type member2;
	
	    data_type memeberN;
	
	};

4. Armstrong number program	#include<stdio.h>
	 int main()
	{
	int n,r,sum=0,temp;
	printf("enter the number=");
	scanf("%d",&n);
	temp=n;
	while(n>0)
	{
	r=n%10;
	sum=sum+(r*r*r);
	n=n/10;
	}
	if(temp==sum)
	printf("armstrong number ");
	else
	printf("not armstrong number");
	return 0;
	}

5. Code to swap 2 numbers without temporary variable.	**Logic:**
	1. Take two integer as input num1 and num2.
	2. Print number before swapping
	3. num1 = num1 + num2;
	4. num2 = num1 - num2;
	5. num1 = num1 - num2;
	6. Print numbers after swapping
	
	Program:
	
	#include <stdio.h>
	int main()
	{
	int num1,num2;
	printf("Enter two numers : \n");
	scanf("%d %d",&num1,&num2);
	printf("Number before swapping is %d and %d \n",num1,num2);
	num1=num1+num2;
	num2=num1-num2;
	num1=num1-num2;
	printf("Number after swapping is %d and %d

	\n",num1,num2);
	return 0;
	}

6. Leap year Program	**Logic:**
	1. Take integer variable year
	2. Assign value to the variable
	3. Check if year is divisible by 4 but not 100, DISPLAY "leap year"
	4. Check if year is divisible by 400, DISPLAY "leap year"
	5. Otherwise, DISPLAY "not leap year"
	
	Program:
	
	#include <stdio.h>
	int main() {
	  int year;
	  year = 2016;
	
	  if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))
	    printf("%d is a leap year", year);
	  else
	    printf("%d is not a leap year", year);
	
	  return 0;
	}

7. Write a program to identify characters, digits, and special characters in a given string	#include <string.h>
	int main()
	{
	   char s[1000];
	   int i,alphabets=0,digits=0,specialcharacters=0;
	   printf("Enter the string : ");
	   gets(s);
	
	   for(i=0;s[i];i++)
	   {
	      if((s[i]>=65 && s[i]<=90)|| (s[i]>=97 && s[i]<=122))
	        alphabets++;
	      else if(s[i]>=48 && s[i]<=57)
	       digits++;
	      else
	       specialcharacters++;
	
	        }
	
	   printf("Alphabets = %d\n",alphabets);
	   printf("Digits = %d\n",digits);
	   printf("Special characters = %d", specialcharacters);

	return 0;
	}

8. How can we convert string into uppercase.	#include <stdio.h>
	#include <string.h>
	int main() {
	  char s[100];
	  int i;
	  printf("\nEnter a string : ");
	  gets(s);
	  for (i = 0; s[i]!='\0'; i++) {
	    if(s[i] >= 'a' && s[i] <= 'z') {
	      s[i] = s[i] -32;
	    }
	  }
	  printf("\nString in Upper Case = %s", s);
	  return 0;
	}

9. How can you generate a random number.	As C does not have an inbuilt function for generating a number in the range, but it does have rand function which generate a random number from 0 to RAND_MAX. With the help of rand () a number in range can be generated as num = (rand() % (upper – lower + 1)) + lower
10. Difference b/w call by value and call by reference.	1. In Call by value method the original value is not modified whereas, in Call by reference method, the original value is modified. 2. In Call by value, a copy of the variable is passed whereas in Call by reference, a variable itself is passed. 3. In Call by value, actual and formal arguments will be created in different memory locations whereas in Call by reference, actual and formal arguments will be created in the same memory location. 4. Call by value is the default method in programming languages like C++, PHP, Visual Basic NET, and C# whereas Call by reference is supported only in the Java language. 5. Call by Value, variables are passed using a straightforward method whereas Call by Reference, pointers are required to store the address of variables.
11. Difference Between Break And Continue	1. The **Break** statement is used to exit from the loop constructs. The **continue** statement is not used to exit from the loop constructs. 2. The **break** statement is usually used with the switch statement, and it can also be used within the while loop, do-while loop, or the for-loop. The

	continue statement is not used with the switch statement, but it can be used within the while loop, do-while loop, or for-loop. 3. When a **break** statement is encountered then the control is exited from the loop construct immediately. When the **continue** statement is encountered then the control automatically passed from the beginning of the loop statement.
12. Memory allocation in c	Memory allocation is a process by which computer programs and services are assigned with physical or virtual memory space. The memory allocation is done either before or at the time of program execution. There are two types of memory allocations: **Static Memory Allocation:** Static Memory is allocated for declared variables by the compiler. The address can be found using the address of operator and can be assigned to a pointer. The memory is allocated during compile time. **Dynamic Memory Allocation:** Memory allocation done at the time of execution(run time) is known as dynamic memory allocation. Functions calloc() and malloc() support allocating dynamic memory. In the Dynamic allocation of memory space is allocated by using these functions when the value is returned by functions and assigned to pointer variables.
13. Null pointer, constructor and destructor	A **null pointer** is a pointer which points nothing. Some uses of the null pointer are: a) To initialize a pointer variable when that pointer variable isn't assigned any valid memory address yet. b) To pass a null pointer to a function argument when we don't want to pass any valid memory address. c) To check for null pointer before accessing any pointer variable. So that, we can perform error handling in pointer related code e.g. dereference pointer variable only if it's not NULL A **Constructor** in C is used in the memory management of C++programming. It allows built-in data types like int, float and user-defined data types such as class. Constructor in Object-oriented programming initializes the variable of a user-defined data type. Constructor helps in the creation of an object. The name of the constructor is the same as the name of the object but it has no return type. A Constructor is executed

<table>
<tr>
<td></td>
<td>

automatically when an object or special member is created. It allocates the memory for the new object created and it can be overloaded.

Destructor is an instance member function which is invoked automatically whenever an object is going to be destroyed. Meaning, a destructor is the last function that is going to be called before an object is destroyed.

</td>
</tr>
<tr>
<td>14. Memory leakage and how to identify</td>
<td>

Memory leak occurs when programmers create a memory in heap and forget to delete it. The consequences of memory leak is that it reduces the performance of the computer by reducing the amount of available memory.

Memory leaks in C happen for three core reasons:

1. we do not free the memory that is no longer needed
2. we do try to free the memory but we do not have the reference to it (dangling pointer)
3. we try to free the memory using the wrong function

Tools like Valgrind & Visual Studio can help to identify the memory leks without any hassle.

</td>
</tr>
<tr>
<td>15. Static variable</td>
<td>

Static variables are initialized only once. The compiler persists with the variable till the end of the program. Static variables can be defined inside or outside the function. They are local to the block. The default value of static variables is zero. The static variables are alive till the execution of the program.

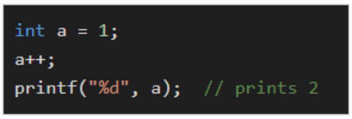
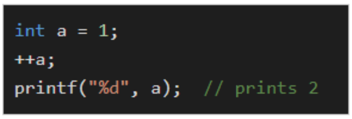
</td>
</tr>
<tr>
<td>16. Functions with example</td>
<td>

A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.

```
/* function returning the max between
two numbers */
int max(int num1, int num2) {

   /* local variable declaration */
   int result;

   if (num1 > num2)
      result = num1;
   else
      result = num2;

   return result;
```

</td>
</tr>
</table>

17. Can For condition be used in a string	Yes For loop can be used iterate through a string, as we know every string ends with ‘/0’ in C, we can start the loop from 0 index and go till we find ‘\0’
18. Logical error and compilation error	**Logical Errors :** On compilation and execution of a program, desired output is not obtained when certain input values are given. These types of errors which provide incorrect output but appear to be error free are called logical errors. These are one of the most common errors done by beginners of programming. **Compilation error:** It refers to a state when a compiler fails to compile a piece of computer program source code, either due to errors in the code, or, more unusually, due to errors in the compiler itself. A compilation error message often helps programmers debugging the source code

19. Difference between ++a and a++(same output or different)	a++ is called the post-increment and the ++a is called the pre-increment. Both of them have different precedence.
	
	int a = 1;
	a++;
	printf("%d", a); // prints 2
	
	int a = 1;
	++a;
	printf("%d", a); // prints 2
	However, the ++ position can make a difference when you are reading the value of the variable in the same statement. More precisely, the post-increment a++ and the pre-increment ++a have different precedence.
	int a = 1;
	int b = a++;
	printf("%d", b); //Prints 1
	int a = 1;
	int b = ++a;
	printf("%d", b); //Prints 2

20. Loops in C	A loop statement allows us to execute a statement or group of statements multiple times. **1. While loop** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body. **2. for loop** Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. **3 do...while loop** It is more like a while statement, except that it tests the

	condition at the end of the loop body. **4. nested loops** You can use one or more loops inside any other while, for, or do..while loop.
21. Difference between do while and while	1. In **While**, Condition is checked first then statement(s) is executed. In **do-while,** Statement(s) is executed at least once, thereafter the condition is checked. 2. In **while** It might occur statement(s) is executed zero times, If the condition is false. In **do-while**, At least once the statement(s) is executed. 3. **while** loop is entry controlled loop. **do-while loop** is exit controlled loop.
22. What is pointer?	A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address.
23. Data types	A data type or simply type is an attribute of data which tells the compiler or interpreter how the programmer intends to use the data. Following are the examples of some very common data types used in C: ● **char**: The most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers. ● **int**: As the name suggests, an int variable is used to store an integer. ● **float:** It is used to store decimal numbers (numbers with floating point value) with single precision. ● **double**: It is used to store decimal numbers (numbers with floating point value) with double precision.
24. Difference between float and double, When to use float and double	1. **Float** has a 32-bit floating point precision. **Double** has 64-bit floating point precision. 2. **Float** takes 4 bytes for storage.**Double** takes 8 bytes for storage. 3. **Float** has a precision of 6 decimal places. **Double** has a precision of 15 decimal places.
25. Calloc and Malloc	**Calloc():** calloc() in C is a function used to allocate multiple blocks of memory having the same size. It is a dynamic memory allocation function that allocates the memory space to complex data structures such as arrays and structures and returns a void pointer to the memory. Calloc stands for contiguous allocation.

	ptr = (cast_type *) calloc (n, size);
	Malloc(): The malloc() function stands for memory allocation. It is a function which is used to allocate a block of memory dynamically. It reserves memory space of specified size and returns the null pointer pointing to the memory location. The pointer returned is usually of type void. It means that we can assign malloc function to any pointer.
	ptr = (cast_type *) malloc (byte_size);

26. Null pointer and double pointer	**NULL POINTER:** A null pointer is a pointer which points nothing. Some uses of the null pointer are: a) To initialize a pointer variable when that pointer variable isn't assigned any valid memory address yet. b) To pass a null pointer to a function argument when we don't want to pass any valid memory address. c) To check for null pointer before accessing any pointer variable. So that, we can perform error handling in pointer related code e.g. dereference pointer variable only if it's not NULL A **double pointer** is used to store the address of variables. So, when we define a pointer to pointer, the first pointer is used to store the address of the second pointer. Thus it is known as double pointers.
27. Storage classes and Static variable	Storage Classes are used to describe the features of a variable/function. These features basically include the scope, visibility and life-time which help us to trace the existence of a particular variable during the runtime of a program. We have four different storage classes in a C program − ● Auto: The auto storage class is the default storage class for all local variables. ● Register: The register storage class is used to define local variables that should be stored in a register instead of RAM for faster accessing. ● Static: The static storage class instructs the compiler to keep a local variable in existence during the life-time of the program instead of creating and destroying it each time it comes into and goes out of scope.

	● Extern: The extern storage class is used to give a reference of a global variable that is visible to ALL the program files. Static variables are initialized only once. The compiler persists with the variable till the end of the program. Static variables can be defined inside or outside the function. They are local to the block. The default value of static variables is zero. The static variables are alive till the execution of the program.
28. Memory leakage?	Memory leak occurs when programmers create a memory in heap and forget to delete it. The consequences of memory leak is that it reduces the performance of the computer by reducing the amount of available memory. Eventually, in the worst case, too much of the available memory may become allocated and all or part of the system or device stops working correctly, the application fails
29. Storage limit of float and double	**Float** takes 4 bytes for storage. **Double** takes 8 bytes for storage. A value having a range within 1.2E-38 to 3.4E+38 can be assigned to **float variables**. A value having range within 2.3E-308 to 1.7E+308 can be assigned to **double** type variables

30. where should we use double pointer?	We use double-pointer when we want to protect the memory allocation or an assignment even outside of a function call.
	To allocate space for a matrix or multi-dimensional arrays dynamically, you will need a double-pointer.
	We can use Pointers to pointers as “handles” to memory where you want to pass around a “handle” between functions to re-locatable memory.
	`syntax:`
	`int **ptr;    // declaring double pointers`
	A double pointer is basically a pointer to pointer.
	A pointer denotes the address of a particular data-value(type maybe int,float,char or some user-defined data types).
	A pointer to pointer is the address of an address of a data.

31. basic data types in c	Basic Data Type-int, char, float, double Derived Data Type-array, pointer, structure, union Enumeration Data Type-enum Void data type- void

	The basic data types are int, char, and float. Int is used to declare variables that will be storing integer values. Float is used to store real numbers. Char can store individual character values.
32. DS and its types	Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. ● Primitive data structure ● Non-primitive data structure The primitive data structures are primitive data types. The int, char, float, double, and pointer are the primitive data structures that can hold a single value. The non-primitive data structure is divided into two types: ● Linear data structure ● Non-linear data structure The arrangement of data in a sequential manner is known as a linear data structure. The data structures used for this purpose are Arrays, Linked list, Stacks, and Queues. When one element is connected to the 'n' number of elements known as a non-linear data structure. The best example is trees and graphs. In this case, the elements are arranged in a random manner.
33. What happens if we change i++ to ++i in for loop	The reason is that ++i and i++ are two different operations. The ++ before 'i' is called a pre-increment operator. And, the ++ after 'i' is called post-increment. For example, consider the code: int i=0;

	print(++i);
	The output of above code will be 1.
	But, consider this code:
	int i=0;
	print(i++);
	The output of this code will be 0.
	The reason is that ++i first increases the value of i by 1 and then prints it. Whereas, i++ first prints the value of i and then increases its value by 1.

34. Dangling Pointer	A pointer pointing to a memory location that has been deleted (or freed) is called a dangling pointer. Dangling pointer occurs at the time of the object destruction when the object is deleted or de-allocated from memory without modifying the value of the pointer. In this case, the pointer is pointing to the memory, which is de-allocated.
35. Compile time and run time	**Compile-time:** ● This is when the code is translated from a programming language to a language that a machine understands. ● These errors are referenced to an error in syntax or semantics. ● This error can be detected during software development. ● These errors can be fixed during the code development itself. **Runtime**: ● This is when a code is run in the runtime environment and starts from the time code execution starts till the point the user or OS stops the code. ● These errors are a reference to the execution of the code in a runtime environment. ● This error might come up even when the code is run in a runtime environment, even if it has passed in the local.

36. Operators used in c language	● An operator is a symbol that tells the compiler to perform a certain mathematical or logical operations, based on the values provided to the operator. ● Operators are used in programs to manipulate data and variables. ● Arithmetic operators ● Relational operators ● Logical operators ● Bitwise operators ● Assignment operators ● Conditional operators ● Special operators
37. Applications of c and java programming languages?	**JAVA APPLICATIONS:** Mobile Applications Desktop GUI Applications Web-based Applications Enterprise Applications Scientific Applications Gaming Applications Big Data technologies Business Applications Distributed Applications Cloud-based Applications **C APPLICATIONS:** Operating Systems Language Compilers Assemblers Text Editors Print Spoolers Network Drivers Modern Programs Data Bases Language Interpreters Utilities
38. what Is library in c and when it will be used.	Library functions in C language are inbuilt functions which are grouped together and placed in a common place called library. The use of the library function is to get the predefined output instead of writing our own code to get those outputs.

	Libraries in two types: static and dynamic. Static library is just a bag of object files — containing functions, while dynamic library is part of runtime.
39. static and global variables	The variables declared outside any function are called global variables. They are not limited to any function. Any function can access and modify global variables. Global variables are automatically initialized to 0 at the time of declaration. Global variables are generally written before main() function. A Static variable is able to retain its value between different function calls. The static variable is only initialized once, if it is not initialized, then it is automatically initialized to 0. Here is how to declare a static variable.

40. Difference between do while and for	**For Loop:**
	For loop Statement(s) is executed once the condition is checked.
	It might be that the statement(s) gets executed zero times.
	Initialization may be outside or in condition box.
	for loop is entry controlled loop.
	for (init ; condition ; iteration)
	{ statement (s); }
	Do-While Loop:
	Condition is checked after the statement(s) is executed.
	Statement(s) is executed at least once.
	Initialization may be outside or within the loop.
	do-while is an exit controlled loop.
	do
	{
	statement(s);
	}while (condition);

41. What is a virtual keyword?	A 'virtual' is a keyword preceding the normal declaration of a function. When the function is made virtual, C++

	determines which function is to be invoked at the runtime based on the type of the object pointed by the base class pointer.
42. Can we use virtual to a function in base class?	Virtual functions cannot be static members. They are accessed through object pointers. They can be a friend of another class. A virtual function must be defined in the base class, even though it is not used.
43. command to come out of loop	
44. what happens if stdio.h is not included	stdio.h is the header file for standard input and output. This is useful for getting the input from the user(Keyboard) and output result text to the monitor(screen). With out this header file, one can not display the results to the users on the screen or cannot input the values through the keyboard
45. use of header file	Header files are most often used to include external variable declarations, macro definitions, type definitions, and function declarations. Header files serve two purposes. ● System header files declare the interfaces to parts of the operating system. You include them in your program to supply the definitions and declarations you need to invoke system calls and libraries. ● Your own header files contain declarations for interfaces between the source files of your program. Each time you have a group of related declarations and macro definitions all or most of which are needed in several different source files, it is a good idea to create a header file for them.
46. what is stack and queue	A Stack is a linear data structure. It is a container that follows the insertion and deletion rule. It follows the principle LIFO (Last In First Out) in which the insertion and deletion take place from one side known as a top. In stack, we can insert the elements of a similar data type, i.e., the different data type elements cannot be inserted in the same stack. The two operations are performed in LIFO, i.e., push and pop operation. A Queue is a linear data structure. It is an ordered list that follows the principle FIFO (First In -First Out). In the case of Queue, insertion is performed from one end, and that end is known as a rear end. The deletion is performed from another end, and that end is known as a front end. In Queue, the technical words for insertion and deletion are enqueue() and dequeue(), respectively

	whereas, in the case of the stack, the technical words for insertion and deletion are push() and pop().
47. what is modular programming	Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality
48. what is differences b/w struct and union	1. Every member within the structure is assigned a unique memory location. In union, a memory location is shared by all the data members. 2. Changing the value of one data member will not affect other data members in structure. Changing the value of one data member will change the value of other data members in union. 3. It enables you to initialize several members at once. It enables you to initialize only the first member of the union. 4. The total size of the structure is the sum of the size of every data member. The total size of the union is the size of the largest data member.
49. what is the use of printf and scanf	printf() is an inbuilt library function in C which is available in C library by default. This function is declared and related macros are defined in “stdio.h” header file. printf() function is used to print the “character, string, float, integer, octal and hexadecimal values” onto the output screen. scanf() function is an inbuilt library function in C which is available in C library by default. This function is declared and related macros are defined in the “stdio.h” header file. scanf() function is used to read character, string, numeric data from the keyboard.
50. Difference b/w %d and %c	A format specifier is used to specify the format in which we need the output to be printed, %d prints the value in decimal format whereas %c prints it in character format.
51. what is variable	C variable is a named location in a memory where a program can manipulate the data. This location is used to hold the value of the variable. The value of the C variable may change in the program. C variable might be belonging to any of the data type like int, float, char etc.

52. what is token	C tokens are the basic building blocks in C language which are constructed together to write a C program. Each and every smallest individual unit in a C program are known as C tokens. Keywords ,Identifiers strings, Operators, Constants ,Special Characters are the tokens.
53. what is Ansi	ANSI is a US standards body that released the first standard specification for C in 1989. The standard was adopted by ISO in 1990 and ISO are now the standards body for the language - not ANSI. ANSI C merely refers to a particular standard for the C Programming Language - i.e. there is no difference, they refer to the same thing. However there have been several standards and de-facto standards for C:
54. Bitwise operation	Bitwise operators are used to perform bit operations. Decimal values are converted into binary values which are the sequence of bits and bit wise operators work on these bits. Bit wise operators in C language are & (bitwise AND), | (bitwise OR), ~ (bitwise OR), ^ (XOR), << (left shift) and >> (right shift).

55. Multiplication and division using bitwise operators	In this case, you can multiply by using left shift and divide by using right shift, that is
	int c = a << m; // multiply a by 2^m
	int d = a >> m; // divide a by 2^m
	This works because the shift operator works in the binary domain and zero-fills (or maintains the sign bit for right-shift), so a left shift is equivalent to multiplying by 2 (and right shift is equivalent to dividing). You can imagine that if these operators lived in the base-10 domain, a zero-filled shift to the left is like multiplying by 10 and vice-versa for a right shift.

56. escape sequence in c	An escape sequence in C language is a sequence of characters that doesn't represent itself when used inside string literal or character.

	It is composed of two or more characters starting with backslash \. For example: \n represents a new line.
57. Delete operator in C	A delete operator is used to deallocate memory space that is dynamically created using the new operator, calloc and malloc() function, etc., We can delete a specific element or variable using the delete operator, as shown: delete pointer_variable; // delete ptr; It deallocates memory for one element
58. Lifo and Fifo	FIFO is an abbreviation for first in, first out. It is a method for handling data structures where the first element is processed first and the newest element is processed last. The data structure that implements FIFO is Queue. LIFO is an abbreviation for Last in, first out is the same as fist in, last out (FILO). It is a method for handling data structures where the last element is processed first and the first element is processed last. The data structure that implements LIFO is Stack.
59. High and Low level Languages	High level languages – These level languages provide almost everything that the programmer might need to do as already built into the language. Example: Java, Python Middle level languages – These languages don’t provide all the built-in functions found in high level languages, but provide all the building blocks that we need to produce the result we want. Example: C, C++ Low level languages – These languages provide nothing other than access to the machine’s basic instruction set. Example: Assembly language.

60. Sorting Techniques	Arranging the data in ascending or descending order is known as sorting. The best example of sorting can be phone numbers in our phones. If they are not maintained in an alphabetical order we would not be able to search any number effectively. 1.Bubble sort 2. Selection sort 3. Insertion sort 4. Quick sort 5. Merge sort 6. Heap sort 7. Radix sort 8. Shell sort
61. Tell about linear and nonlinear	**Linear:** In this structure, the elements are arranged sequentially or linearly and attached to one another. Arrays, linked list, stack, queue are the types of a linear data structure In this, the memory utilization is not efficient. **Non-Linear:** In this structure, the elements are arranged hierarchically or in a non-linear manner. Trees and graphs are the types of a non-linear data structure. In this, memory is utilized in a very efficient manner.
62. Tree is linear or nonlinear	In the tree we can approach more than one data item . So we can say that trees are non-linear data structures.

	A non-linear data structure is a data structure in which a data item is connected to several other data items. So that a given data item has the possibility to reach one-or-more data items.

DBMS
Video Link: https://drive.google.com/file/d/1QVaXDd4TjfqSJqWD_GbftEkRSji3K13d/view?usp=sharing

Question	Answer
1. What is a database?	A database is a collection of interrelated data that belongs to a particular organization. This data will have some inherent meaning, representing some aspect of the real world and is designed for a specific purpose.
2. Query to find second highest salary in employee table	Select max(salary) From employee Where salary < (select max(salary) from employee) **Explanation:** In the above query, the inner query returns the first highest salary. In the outer query, we are trying to find the maximum salary that is less than the first highest salary (which is second highest salary)
3. Primary key and foreign key, Different kinds of keys, is primary key necessary.	**Primary Key:** Primary key (of a table) is a set of attributes whose values are UNIQUE & NOT NULL. The primary key is used to identify records uniquely from a table. **Foreign key** (of a table) is a set of attributes that identifies the records in another table (parent table). **Example**: EMP(**eid**, ename, age, sal, **did**) DEPT(**did**, dname, location, budget) In the above example, the primary key for EMP table is eid. The primary key for the DEPT table is did. The did in the EMP table is the foreign key & this foreign key is referencing the table DEPT (parent table). All the values that appear under did in EMP table should be present in did column in the DEPT table.

	Different kinds of KEYS: Different kinds of KEYS are: ● Candidate Key ● Super Key ● Primary Key ● Alternate Key ● Foreign Key Candidate key is the minimal set of attributes that identify the records uniquely from a table Super key is the set of attributes that identify the records uniquely from a table Primary key & Alternate Keys: One of the candidate keys chosen by Database Administrator (DBA) as the key for the table is called Primary Key. A given table may have more than one candidate key. In that case, the key chosen by the DBA to identify records from the table is called Primary key. And the remaining candidate keys become Alternate Keys. Foreign key (of a table) is a set of attributes that identifies the records in another table (parent table). **Is Primary Key necessary?** We can create a table without a Primary Key also. But, creating Primary Key will ensure the integrity of the data in the table (i.e. ensures no duplicates & no missing values)
4. DDL, DML, DCL commands	**DDL:** These are the SQL commands that work with table definitions like creating tables, altering table definitions, deleting table definitions. Example: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE **DML**: These are the SQL commands that work with the data in the tables (manipulate the data in the tables). Example: INSERT, DELETE, UPDATE, SELECT **DCL:** These are the SQL commands used to grant & revoke permissions on database objects. Example: GRANT, REVOKE
5. SQL joins and questions based on them? (Most common question)	An SQL Join is used to combine records from multiple tables. **Different types of Joins in SQL:** → Natural Join → Inner Join → Self Join → Equi Join → Outer Join (left outer join, right outer join, full outer join)

Natural join is a join operation which enforces equality on common attributes between the tables.
Example:
EMP(eid, name, age, did, sal)
DEPT(did, dname, location)

Write an SQL query to find the employee names and their department names

```
SELECT name, dname
FROM  emp NATURAL JOIN dept
```

Inner Join is a join operation, which joins two tables according to the matching condition.

Example1:

```
SELECT *
FROM  R INNER JOIN S
ON  R.C=S.C;
```

Example2:

```
SELECT *
FROM R INNER JOIN S
ON R.C>S.C;
```

Self Join: Joining a table with itself is called self join.

Example:

EMP(eid, ename, mgrid)

Find the names of employees & their managers

```
select  e1.ename  empname,  e2.ename  mgrname

from  emp  e1, emp  e2

where e1.mgrid = e2.eid;
```

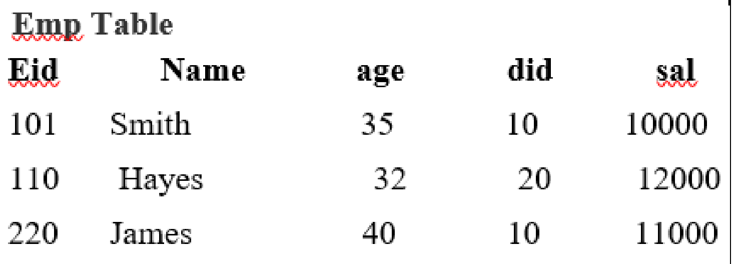
Equi join is a join operation where the join condition consists of only equality.

Outer Join is a join operation that returns all rows from both the participating tables which satisfy the join condition along with rows which do not satisfy the join condition.

Various Outer Joins:

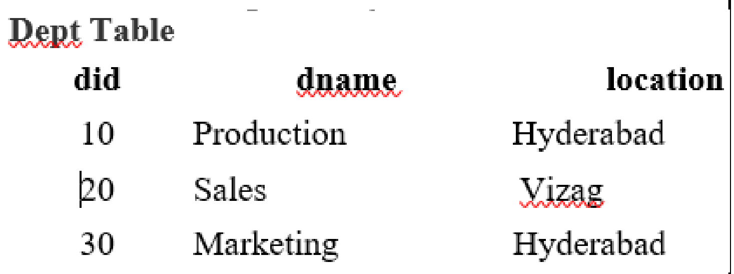
LEFT OUTER JOIN or LEFT JOIN
RIGHT OUTER JOIN or RIGHT JOIN
FULL OUTER JOIN

Example:



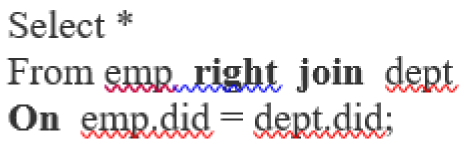
Emp Table

Eid	Name	age	did	sal
101	Smith	35	10	10000
110	Hayes	32	20	12000
220	James	40	10	11000

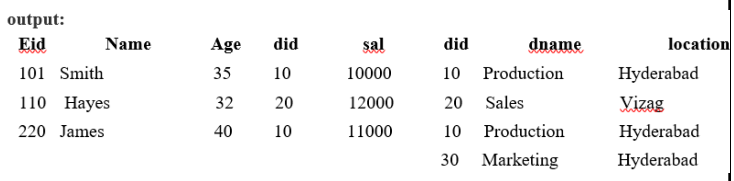


Dept Table

did	dname	location
10	Production	Hyderabad
20	Sales	Vizag
30	Marketing	Hyderabad



```
Select *
From emp right join dept
On emp.did = dept.did;
```



output:

Eid	Name	Age	did	sal	did	dname	location
101	Smith	35	10	10000	10	Production	Hyderabad
110	Hayes	32	20	12000	20	Sales	Vizag
220	James	40	10	11000	10	Production	Hyderabad
					30	Marketing	Hyderabad

In the above example, as the right outer join is used, for the records in right side table (i.e. dept) for which the matching condition is not met, those records are also appended to the output.

<u>Use of outer join</u>

Outer join is used to deal with missing information.

As shown in the above example, we want to display

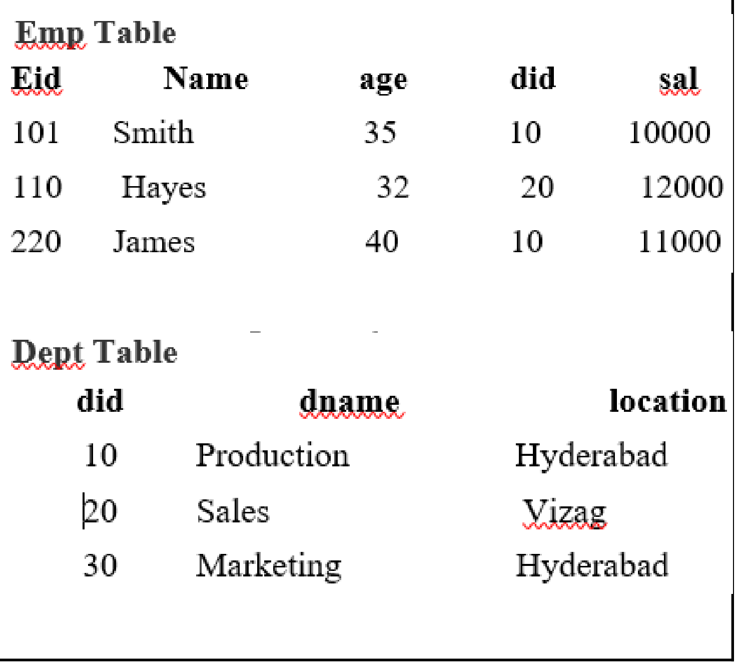
	emp & their department information. Even if there are no employees working in any department, that department details should also be displayed, if that is our requirement, then we want write the query as shown above.
6. What are null values in the table?	NULL values indicate the absence of information about the value of an attribute. It is used when we don't know attribute value or not applicable.
7. Delete, drop and Truncate	DELETE is a DML command that is used to delete the records that satisfy the given condition. When no condition is specified, it deletes all the records from the table. As DELETE is a DML command, the action of DELETE can be rolled back. Syntax: DELETE FROM <tablename> WHERE <cond>; TRUNCATE is a DDL command, It deletes all the records from the table, As TRUNCATE is a DDL command, its effect can't be rolled back. Syntax: TRUNCATE TABLE <tablename>; DROP is a DDL command that deletes the table structure (so table along with the data will be lost). Syntax: DROP TABLE <tablename>;
8. Languages in dbms	The various languages in DBMS are: ● DDL - Data Definition Language ● DML - Data Manipulation Language ● DCL - Data Control Language ● TCL - Transaction Control Language DDL: These are the SQL commands that work with table definitions like creating tables, altering table definitions, deleting table definitions. Example: CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE DML: These are the SQL commands that work with the data in the tables (manipulate the data in the tables). Example: INSERT, DELETE, UPDATE, SELECT DCL: These are the SQL commands used to grant & revoke permissions on database objects. Example: GRANT, REVOKE TCL: These are the SQL commands used to control the actions of transactions.

	Example: COMMIT, ROLLBACK
9. Sub queries, Nested Query	Nested Query: Query within another Query is called Nested Query. A subquery is a query that is nested inside a SELECT , INSERT , UPDATE , or DELETE statement, or inside another subquery. A subquery can be used anywhere an expression is allowed. Example: EMP(eid, name, age, did, sal) DEPT(did, dname, location) Find the names of employees working in location "Hyderabad" SELECT name FROM emp WHERE did IN (SELECT did FROM dept WHERE location = 'Hyderabad')
10. what is a transaction?	Execution of a user program in a database environment is called a Transaction. Transaction will have 4 properties (ACID properties) A - Atomicity C - Consistency I - Isolation D - Durability

11. Foriegn key syntax	CREATE TABLE <tablename>(column1 datatype, column2 datatype,...., foreign key(column) references <parenttable>);
	Example:
	EMP(**eid**, ename, age, sal, **did**)
	DEPT(**did**, dname, location, budget)
	Create table dept(did number(3) PRIMARY KEY, dname varchar2(30), location varchar2(30), budget number(8));
	Create table emp(eid number(3) PRIMARY KEY, ename varchar2(30), age number(2), sal number(6), did number(3) FOREIGN KEY REFERENCES dept);
	In the above example, DEPT is the parent table & EMP is the child table.
	The did in the EMP table is the foreign key & this foreign key is referencing the table DEPT (parent

	table). RULES to be followed while creating FOREIGN KEY: 1) First the parent table should be created 2) The foreign key in the child table will be the primary key in the parent table.

12. Query to find employee name ends between a to s	SELET *
	FROM emp
	WHERE name LIKE ‘%[a-s]’

13. Surrogate key	Surrogate key, also called a synthetic primary key, is generated when a new record is inserted into a table automatically by a database that can be declared as the primary key of that table . In case we do not have a natural primary key in a table, then we need to artificially create one in order to uniquely identify a row in the table. This key is called the surrogate key or synthetic primary key of the table.
14. Indexing	Indexing technique helps to retrieve the records efficiently in database systems. In database systems, B Trees and B+ Trees are used to create index data structures. Syntax to create Index: CREATE INDEX <indexname> ON <tablename>(columnname); Example: CREATE INDEX emp_ind ON emp(eid);
15. Write a left outer join query with example	

Emp Table

Eid	Name	age	did	sal
101	Smith	35	10	10000
110	Hayes	32	20	12000
220	James	40	10	11000

Dept Table

did	dname	location
10	Production	Hyderabad
20	Sales	Vizag
30	Marketing	Hyderabad

	Example: Write an SQL query to display department details along with the employees working in those departments. If there are no employees working in a particular department, that department information should also be displayed in the output.
	
	SELECT *
	FROM Dept LEFT JOIN Emp
	ON Dept.did = Emp.did

16. equi join outer join full outer join	**Equi join** is a join operation where the join condition consists of only equality. **Outer Join** is a join operation that returns all rows from both the participating tables which satisfy the join condition along with rows which do not satisfy the join condition. **Various Outer Joins:** LEFT OUTER JOIN or LEFT JOIN RIGHT OUTER JOIN or RIGHT JOIN FULL OUTER JOIN
17. ACID properties	ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably. **Atomicity** – Either all actions of the transaction should be done or none of the actions should be done **Consistency** - The consistency property ensures that the database remains in a consistent state before the start of the transaction and after the transaction is over (whether successful or not). The consistency property ensures that any transaction will bring the database from one valid state to another. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof. **Isolation** – The isolation property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other. Providing isolation is the main goal of concurrency control.

	Durability – If a transaction has been completed successfully, its results should persist despite system failures.
18. Difference between procedure and function	The function **must return a value** but in Stored Procedure it is optional. Even a procedure can return zero or n values.
19. what is a table?	Table (relation) is a collection of data which is organized in the form of rows and columns. In relational database, the data is organized in the form of tables.
20. what is view?	View is a ***virtual table*** in which the records are not stored, but only the view definition is stored. When we access the view, the result of the view will be computed from its definition.
21. what is stored procedure?	Stored procedure is a batch of statements grouped as a logical unit and stored in the database. When the stored procedure is invoked, it is executed and the result set is returned.
22. what is index and types of index.	Index is a data structure that is used to retrieve the records efficiently from the database. Different types of indexes are tree index, hash index. In database systems, tree indexes (B Tree, B+ Tree) are used.

23. asked syntax of updating salary of an employee in employee table.	Syntax:
	UPDATE <tablename>
	SET col1 = exp1, col2 = exp2, ..
	[WHERE <condition>];
	
	Example:
	UPDATE employee
	SET salary = salary * 1.1
	WHERE did=10;
	
	The above query will give 10% increment in the salary to the employees working in did=10

24. Sql vs noSql	SQL databases are relational, NoSQL databases are non-relational. SQL databases use structured query language and have a predefined schema. NoSQL databases have dynamic schemas for unstructured data. Examples for SQL databases: Oracle, MySQL Examples for NoSQL databases: MongoDB, Cassandra

25. What is plsql	PL/SQL stands for Procedural Language Structured Query Language. SQL doesn’t have procedural language constructs like loops etc. PL/SQL provides Procedural Language capabilities to SQL.

26. How can we get the unique elements from table	Syntax:
	SELECT DISTINCT col1, col2, col3..
	FROM <tablename>
	
	Example:
	SELECT DISTINCT ename
	FROM employee

27. Aggregate functions	Aggregate functions take a collection of values as input and produce a single value as output. Various aggregate functions in SQL are: SUM, COUNT, AVG, MIN, MAX Ex1: Average sal of all emp SELECT avg(sal) From emp
28. Cursor	A cursor is a temporary work area created in the system memory when a SQL statement is executed. This temporary work area is used to store the data retrieved from the database, and manipulate this data.
29. What is materialized view in database	**Materialized view is a database object that contains the results of a query.** For example, if can be a local copy of data located remotely, or may be a subset of rows and/or columns of a table or join result, or may be a summary using an aggregate function. **Materialized views improve performance.** Example: Consider a distributed database, where the database is spread across various sites. Consider a query at site-1, accessing a portion of data from site-2 frequently, then the copy of the portion of data can be maintained at site-1. This is known as materialized views. This will improve the performance of the system.
30. Normalization and denormalization	Normalization is the process of refining database tables by using decomposition technique. It is used to achieve the following desirable properties: 1) Minimizing Redundancy

	2) Minimizing the Insertion, Deletion, And Update Anomalies. Denormalization is the reverse of normalization, wherein the database tables are combined into a bigger table. It is generally used to improve performance.

LINK TO THE ADDITIONAL MATERIAL ON DBMS INTERVIEW QUESTIONS:
https://drive.google.com/file/d/1FYZhgk7jbFChYDxcKqVcj1rJXtjzA0Og/view?usp=sharing

Note: Use vishnu logins to access the above document (DBMS additional material)

OS
Video Link : https://drive.google.com/file/d/1ySP8iB8ELFS9v34JMq69nJdre0Y-p5oK/view?usp=sharing

<table>
<tr><th>Question</th><th>Answer</th></tr>
<tr><td>1. What is an Operating system?</td><td>Operating system is an interface between the user and the computer hardware.</td></tr>
<tr><td>2. What is the first operating system in a computer and what are the operating systems you know?</td><td>GM-NAA (<b>G</b>eneral <b>M</b>otors and <b>N</b>orth <b>A</b>merican <b>A</b>viation) I/O was the first OS for the IBM 704 computer.
<b>Known OS's are:</b> Windows, Linux, Ubuntu, MAC, Android, BADA (used for Samsung Mobile Phones), Symbian (used for Nokia Mobile Phones), and Apple iOS.</td></tr>
<tr><td>3. What is the difference between multitasking and multiprocessing?</td><td><b>Multi Tasking:</b> Performs more than one task at a time using a single <b>processor</b>. It requires more time to execute the tasks.
<b>Multi Processing:</b> Performs more than one task at a time using <b>Multiple Processors</b>. It requires less time to execute the tasks.</td></tr>
<tr><td>4. Differences between Kernel and OS?</td><td><table><tr><th>Kernel</th><th>OS</th></tr><tr><td>It simply acts as an</td><td>It simply acts as an</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>interface between hardware and applications.</td><td>interface between hardware and user.</td></tr>
<tr><td>It is considered a central component of the OS.</td><td>It is considered system software.</td></tr>
<tr><td>Responsible for converting user commands into machine-level commands.</td><td>Responsible for managing the resources of the system.</td></tr>
</table></td></tr>
<tr><td>5. What is the difference between a program and a process?</td><td>Program is a Passive entity and a process is an active entity.</td></tr>
<tr><td>6. What is processor scheduling?</td><td>CPU takes the process that is ready to execute and allocates CPU to one of them is called processor scheduling.</td></tr>
<tr><td>7.What is Non Preemptive and Preemptive scheduling?</td><td><b>Non Preemptive:</b> Process which is allocated to the CPU will continue to run until it terminates or it gets blocked due to an I/O request.
<b>Pre-emptive:</b> Process which is currently running can be removed from the running state in order to allow another process to run.</td></tr>
<tr><td>8.What is context switching?</td><td>It is a process of saving the context of one process and loading the context of another process. Context switching can occur in multiple processes to share a single CPU.</td></tr>
<tr><td>9. What are the different IPC (Inter Process Communication) Techniques?</td><td>Message Passing, Synchronization, Shared Memory, Remote Procedure Call.</td></tr>
<tr><td>10.What is starvation? Where will it happen?</td><td>An Infinite amount of blocking a process is called starvation. When we use Priority and SJF scheduling, starvation can happen.</td></tr>
<tr><td>11. What is Deadlock?</td><td>It is a situation where a set of processes are blocked as each process is holding resources and waits to acquire resources held by another process.</td></tr>
</table>

12. What are the necessary conditions for Deadlock?	Mutual Exclusion, Hold and Wait, Circular Wait and No Pre-emption.
13. How to identify a Deadlock?	With the help of Resource Allocation Graph(RAG) we can identify the Deadlock.
14. What is a Semaphore? Why is it used?	Semaphore is a signaling mechanism. Used to solve the issue of critical sections in synchronization process using 2 atomic operations ie; wait(), signal().
15.What is a Page Fault?	A page fault occurs when a page referenced by the CPU is not found in the main memory.
16.What is page replacement?	Page replacement is a process of swapping out an existing page from the frame of a main memory and replacing it with the required page.
17. What is Virtual memory?	Virtual memory is a feature of an operating system that enables a computer to be able to compensate for shortages of **physical memory by transferring pages of data from random access memory to disk storage**.
18. What is External and Internal fragmentation?	**External Fragmentation** – total memory space exists to satisfy a request, but it is not contiguous. **Internal Fragmentation** – allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used.
19. How to eliminate external fragmentation?	With the help of a **compaction technique** we can eliminate the external fragmentation.
20.What is Thrashing?	Thrashing is a condition in which excessive paging operations are taking place.

OOPS
Video Link :

Question	**Answer**
1. OOPS concepts with real life examples	

2. Encapsulation example	
3. polymorphism example	
4. Features of oops	
5. Difference between oops concepts in java and oops concepts in c++, is they similar or not?	
6. What is polymorphism and its types? With examples.	
7. Abstraction with example	
8. What is constructor	
9. What is destructor	
10. What is data binding	
11. Inheritance types	
12. What is data abstraction with real world examples	
13. What is an object ?	
14. difference between encapsulation and inheritance	
15. What is dynamic polymorphism	
16. Difference between runtime polymorphism and compile time polymorphism	

JAVA

Video Link :

Question	Answer
1. Applications of c and java programming languages?	
2. Interface and classes differences	
3. What is garbage Collector Thread?	
4. What is static block?	
5. Static binding and dynamic binding?	
6. Oops concepts	
7. Constructors	

8. which is platform independent	
9. Abstract	
10. Exception handle-try, finally	
11. Can we make a constructor final?	
12. Can we have an interface in a class? If so, what do we call it?	
13. Asked about null pointer exception	
14. What is an array?	
15. How can you delete an element in an array and what happens to the array after deleting any two random elements?	
16. How can you delete a character in a character Array?	
17. Jre jdk jvm	
18. Arraylist vs vector	
19. Interfaces, abstract classes, final , static, threads	
20. What is JIT	
21. Explain about class loaders	
22. What is exception handling in java	
23. Difference between throw and throws	
24. Difference between checked and unchecked exceptions	
25. Difference between final finally finalize?	
26. Collections and different classes in collections?	
27. Hashmap and Linkedlist	
28. finally block in exception handling	
29. Static and Super Keyword in java	
30. why java does not support multiple inheritance	
31. Any java frameworks do you know	
32. Differences between python and Java	
33. super class	

34. Wrapper classes	
35. Constructor and constructor chaining	
36. Final and static	
37. User defined exception	
38. Synchronization in java	
39. Public vs private	
40. Serialisation and deserialization	
41. Threading in java	
42. Maps in java	
43. Strings in java	
44. Interpreter vs Compiler	
45. Questions on HashMap Internal working	
46. Diff b/w stack memory and heap memory	
47. what is modular programming	
48. How to implement thread programming language(questions on threads)	
49. difference between method overriding and method overriding.	
50. Java program- class implementation with constructor,one method	
51. How will you remove duplicates from an array	
52. Is a Java program works without mentioning main function as public and why?	
53. Difference between class and abstract class	

HTML
Video Link :

Question	Answer
Tags in HTML	
How to include different type of links in html	

different html tags	
hyperlinks in html	
stylesheet	
HTTP methods	