

# Wipro Technical Interview Questions

## 1.Memory management in C

The C programming language manages memory statically, automatically, or dynamically.

Static-duration variables are allocated in main memory, usually along with the executable code of the program, and persist for the lifetime of the program

Automatic-duration variables are allocated on the stack and come and go as functions are called and return.

For static-duration and automatic-duration variables, the size of the allocation is required to be compile-time constant.

Dynamic memory allocation in which memory is more explicitly (but more flexibly) managed, typically, by allocating it from the heap, an area of memory structured for this purpose.

In C, the library function malloc is used to allocate a block of memory on the heap. The program accesses this block of memory via a pointer that malloc returns. When the memory is no longer needed, the pointer is passed to free which deallocates the memory so that it can be used for other purposes.

## 2.Functionality of Operating System?

An operating system (OS) is a set of software that manages computer hardware resources and provides common services for computer programs.

To act as interface between hardware and users, an operating system must be able perform the following functions:

1. Enabling startup application programs. Thus, the operating system must have:

- A text editor
- A translator
- An editor of links

2. The allocation of resources needed to execute programs is done by identifying: the programs that are running, the need for memory, peripheral devices and data protection requirements.

3. Facilities for data compression, sorting, mixing, cataloging and maintenance of libraries, through utility programs available.

4. Plan implementation works according to certain criteria, for efficient use of central processing unit.

5. Assisting implementation of programs through computer-user communication system, at both hardware and software level.

Examples of operating systems:BS2000,BS3000,DOS,PC-DOS,MS-DOS,LINUX,SOLARIS,MAC OS,UNIX,WINDOWS

### **3.What the use of IP address**

An Internet Protocol address (IP address) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer

network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing.

#### **4.What is difference between UNIQUE and PRIMARY KEY constraints?**

A UNIQUE constraint is similar to PRIMARY key, but you can have more than one UNIQUE constraint per table. Contrary to PRIMARY key UNIQUE constraints can accept NULL but just once. If the constraint is defined in a combination of fields, then every field can accept NULL and can have some values on them, as long as the combination values is unique.

#### **5.What are the steps involved in designing?**

Project plan, Requirements, Design, Coding, Testing, Re-coding and design, Development, Maintenance.

#### **6.what is the difference between interface and multiple interface?**

Both an abstract class and an interface are specific types of computer objects that allow a programmer to loosely define one type of object as if it were another type, while retaining all of the object's original properties. While multiple different computer languages use one or both of these concepts, Java is the most well-known. Abstract classes and interfaces have a variety of similarities, but also incorporate significant differences in structure, syntax, and usage.

#### **7.How can we delete Duplicate row in table?**

```
SQL> delete from table_name where rowid not in (select max(rowid) from table
group by duplicate_values_field_name);
```

#### **8.When do you use SQL Profiler?**

SQL Profiler utility allows us to basically track connections to the SQL Server and also determine activities such as which SQL Scripts are running, failed jobs etc

## **9.What do you meant by active and passive objects?**

Active objects are one which instigate an interaction which owns a thread and they are responsible for handling control to other objects. In simple words it can be referred as client.

Passive objects are one, which passively waits for the message to be processed. It waits for another object that requires its services. In simple words it can be referred as server.

## **10.What do you meant by static and dynamic modeling?**

Static modeling is used to specify structure of the objects that exist in the problem domain. These are expressed using class, object and USECASE diagrams.

But Dynamic modeling refers representing the object interactions during runtime. It is represented by sequence, activity, collaboration and statechart diagrams..

## **11.What is Program counter?**

Program counter holds the address of either the first byte of the next instruction to be fetched for execution or the address of the next byte of a multi byte instruction, which has not been completely fetched. In both the cases it gets incremented automatically one by one as the instruction bytes get fetched. Also Program register keeps the address of the next instruction

## **12.Can you give an example of Stored Procedure?**

CREATE procedure - is a stored procedure, which is a saved collection of Transact-SQL statements that can take and return user-supplied parameters.

## **13.Benefits of Stored Procedures?**

Reduced client/server traffic

Efficient reuse of code and programming abstraction

Enhanced security controls.

**14.Is XML case-sensitive?**

XML is case sensitive when uppercase and lowercase characters are treated differently.

Element type names, Attribute names, Attribute values, All general and parameter entity names, and data content (text), are case-sensitive

**15.What is a Null object?**

It is an object of some class whose purpose is to indicate that a real object of that class does not exist. One common use for a null object is a return value from a member function that is supposed to return an object with some specified properties but cannot find such an object.

**16.What is the property of class?**

A property is a member that provides access to an attribute of an object or a class. Examples of properties include the length of a string, the size of a font, the caption of a window, the name of a customer, and so on.

**17.Does a class inherit the constructors of its super class?**

A class does not inherit constructors from any of its super classes.

**18.If a class is declared without any access modifiers, where may the class be accessed?**

A class that is declared without any access modifiers is said to have package access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package

**19.What do you mean by Stack unwinding?**

It is a process during exception handling when the destructor is called for all local objects between the place where the exception was thrown and where it is caught.

**20.Define precondition and post-condition to a member function.**

**Precondition:** A condition that should return true when a member function is invoked. In order to use a function correctly a precondition should return true. If a precondition fails to hold, an operation will not take responsibility to perform any action of sensibility. For example, the interface invariants of stack class respond nothing about pushing even though the stack is already full. In this scenario, `sinful ()` is a precondition for push operation.

**Post-Condition:** A condition that should return true before returning from an invoked function. In order to use a function correctly a post condition should return true. Taking a stack as an example, `is empty ()` must necessarily be true after pushing the element into the stack when an element is pushed. The function `is empty ()` is a post condition..

## **21.How can you sort the elements of the array in descending order?**

### Syntax

`B = sort(A)`

`B = sort(A,dim)`

`B = sort(...,mode)`

`[B,IX] = sort(A,...)`

### Description

`B = sort(A)` sorts the elements along different dimensions of an array, and arranges those elements in ascending order.

If A is a ...

`sort(A) ...`

Vector

Sorts the elements of A.

Matrix

Sorts each column of A.

Multidimensional array  
returns an array of sorted vectors.

Sorts A along the first non-singleton dimension, and

Cell array of strings                      Sorts the strings in ascending ASCII dictionary order, and returns a vector cell array of strings. The sort is case-sensitive; uppercase letters appear in the output before lowercase. You cannot use the dim or mode options with a cell array.

## **22.Sort - Sort array elements in ascending or descending order**

Integer, floating-point, logical, and character arrays are permitted.

Floating-point arrays can be complex. For elements of A with identical values, the order of these elements is preserved in the sorted list.

When A is complex, the elements are sorted by magnitude, i.e.,  $\text{abs}(A)$ , and where magnitudes are equal, further sorted by phase angle, i.e.,  $\text{angle}(A)$ , on the interval  $[-\pi, \pi]$ . If A includes any NaN elements, sort places these at the high end.

`B = sort(A,dim)` sorts the elements along the dimension of A specified by a scalar dim.

`B = sort(...,mode)` sorts the elements in the specified direction, depending on the value of mode.

'ascend'

Ascending order (default)

'descend'

Descending order

`[B,IX] = sort(A,...)` also returns an array of indices IX, where  $\text{size}(IX) == \text{size}(A)$ . If

A is a vector,  $B = A(IX)$ . If A is an m-by-n matrix, then each column of IX is a permutation vector of the corresponding column of A, such that

for  $j = 1:n$

$B(:,j) = A(IX(:,j),j);$

end

If A has repeated elements of equal value, the returned indices preserve the original ordering.

Example:Sort horizontal vector A:

```
A = [78 23 10 100 45 5 6];
```

```
sort(A)
```

```
ans =5 6 10 23 45 78 100
```

### **23.What is DOM?**

The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents.[1] Objects in the DOM tree may be addressed and manipulated by using methods on the objects. The public interface of a DOM is specified in its application programming interface (API).

### **24.How macro execution is faster than function ?**

Difference between overloading and overriding in programming language is:

- a) In overloading, there is a relationship between methods available in the same class whereas in overriding, there is relationship between a superclass method and subclass method.
- b) Overloading does not block inheritance from the superclass whereas overriding blocks inheritance from the superclass.
- c) In overloading, separate methods share the same name whereas in overriding, subclass method replaces the superclass.
- d) Overloading must have different method signatures whereas overriding must have same signature.

19.what do you mean by realization in oops, what is persistent, transient object.

### **25.Name the operators that cannot be overloaded.?**



There are 5 operators which cannot be overloaded. They are:

. \* - class member access operator

:: - scope resolution operator

. - dot operator

? :: - conditional operator

Sizeof() - operator

Note:- This is possible only in C++

## **26.What is polymorphism?**

In programming languages, polymorphism means that some code or operations or objects behave differently in different contexts.

For example, the + (plus) operator in C++:

4 + 5 <-- integer addition

3.14 + 2.0 <-- floating point addition

s1 + "bar" <-- string concatenation!

In C++, that type of polymorphism is called overloading.

Typically, when the term polymorphism is used with C++, however, it refers to using virtual methods, which we'll discuss shortly.

## **27.What are the differences between a C++ struct and C++ class?**

The default member and base class access specifiers are different.

The C++ struct has all the features of the class. The only differences are that a struct defaults to public member access and public base class inheritance, and a class defaults to the private access specifier and private base class inheritance

## **28.Before interview Please Refer this following programming Questions**

Write a Program for :

1.palindrome for string and number

2.String Reverse

3.Sum,Average of all the number

4.Prime no

5.Armstrong no

6.fibonacci

7.factorial

8.prime number,.

## **29.Palindrome for string**

```
#include <stdio.h>
```

```
#include <string.h>
```

```
main()
```

```
{
```

```
char a[100], b[100];
```

```
printf("Enter the string to check if it is a palindrome\n");
```

```
gets(a);
```

```
strcpy(b,a);
```

```
strrev(b);
```

```
if( strcmp(a,b) == 0 )
```

```
printf("Entered string is a palindrome.\n");
```

```
else
printf("Entered string is not a palindrome.\n");

return 0;
}
```

### **30.Palindrome number in c**

```
#include <stdio.h>
```

```
main()
{
int n, reverse = 0, temp;

printf("Enter a number to check if it is a palindrome or not\n");
scanf("%d",&n);

temp = n;

while( temp != 0 )
{
reverse = reverse * 10;
reverse = reverse + temp%10;
temp = temp/10;
}

if ( n == reverse )
printf("%d is a palindrome number.\n", n);
else
printf("%d is not a palindrome number.\n", n);

return 0;
```

### **31.Reverse a string using C programming**

```
#include<stdio.h>
#include<string.h>
```

```
main()
{
char arr[100];

printf("Enter a string to reverse\n");
gets(arr);
```

```

strrev(arr);

printf("Reverse of entered string is \n%s\n",arr);

return 0;
}

/* Fibonacci Series c language */

#include<stdio.h>

main()
{
int n, first = 0, second = 1, next, c;

printf("Enter the number of terms\n");
scanf("%d",&n);

printf("First %d terms of Fibonacci series are :-\n",n);

for ( c = 0 ; c < n ; c++ )
{
if ( c <= 1 )
next = c;
else
{
next = first + second;
first = second;
second = next;
}
printf("%d\n",next);
}

return 0;
}

```

### **Fibonacci series program in c using recursion**

```

#include<stdio.h>

int Fibonacci(int);

```

```

main()
{
int n, i = 0, c;

scanf("%d",&n);

printf("Fibonacci series\n");

for ( c = 1 ; c <= n ; c++ )
{
printf("%d\n", Fibonacci(i));
i++;
}

return 0;
}

int Fibonacci(int n)
{
if ( n == 0 )
return 0;
else if ( n == 1 )
return 1;
else
return ( Fibonacci(n-1) + Fibonacci(n-2) );
}

```

### **Adding numbers in c using function**

```

#include<stdio.h>

long addition(long, long);

main()
{
long first, second, sum;

scanf("%ld%ld", &first, &second);

sum = addition(first, second);

printf("%ld\n", sum);

return 0;
}

```

```
long addition(long a, long b)
{
    long result;

    result = a + b;

    return result;
}
```