IBS Technical Interview Questions updated on Jan 2025

IBS Technical Interview Questions

IBS Technical interview Questions mainly from C,C++ and DBMS. In Technical interview, be prepared about your favorite subject . Suppose they asked, Which is your favorite Subject/Programming language ? you can easily answer. So be Aware About that....

what you know about c++?
Released in 1985, C++ is an object-oriented programming language created by Bjarne Stroustrup. C++ maintains almost all aspects of the C language, while simplifying memory management and adding several features - including a new datatype known as a class (you will learn more about these later) - to allow object-oriented programming. C++ maintains the features of C which allowed for low-level memory access but also gives the programmer new tools to simplify memory management.
C++ used for:
C++ is a powerful general-purpose programming language. It can be used to create small programs or large applications. It can be used to make CGI scripts or console-only DOS programs. C++ allows you to create programs to do almost anything you need to do. The creator of C++, Bjarne Stroustrup, has put together a partial list of applications written in C++.

What is an object?

Object is a software bundle of variables and related methods. Objects have state and behavior.

What do you mean by inheritance?

Inheritance is the process of creating new classes, called derived classes, from existing classes or base classes. The derived class inherits all the capabilities of the base class, but can add embellishments and refinements of its own.

What is polymorphism?

Polymorphism is the idea that a base class can be inherited by several classes. A base class pointer can point to its child class and a base class array can store different child class objects.

44. What is a scope resolution operator?
A scope resolution operator (::),can be used to define the member functions of a class outside the class.

Anything wrong with this code?

```
T *p = new T[10];
delete p;
```

Everything is correct, Only the first element of the array will be deleted⍰⍰, The entire array will be deleted, but only the first element destructor will be called.

What is Boyce Codd Normal form?

A relation schema R is in BCNF with respect to a set F of functional dependencies if for all functional dependencies in F+ of the form a-> , where a and b is a subset of R, at least one of the following holds:
* a- > b is a trivial functional dependency (b is a subset of a)
* a is a superkey for schema R

What is virtual class and friend class?

Friend classes are used when two or more classes are designed to work together and need access to each other's implementation in ways that the rest of the world shouldn't be allowed to have. In other words, they help keep private things private. For instance, it may be desirable for class DatabaseCursor to have more privilege to the internals of class Database than main() has.

How do you find out if a linked-list has an end? (i.e. the list is not a cycle)

You can find out by using 2 pointers. One of them goes 2 nodes each time. The second one goes at 1 nodes each time. If there is a cycle, the one that goes 2 nodes each time will eventually meet the one that goes slower.
If that is the case, then you will know the linked-list is a cycle.

What is the difference between realloc() and free()?

The free subroutine frees a block of memory previously allocated by the malloc subroutine. Undefined results occur if the Pointer parameter is not a valid pointer. If the Pointer parameter is a null value, no action will occur. The realloc subroutine changes the size of the block of memory pointed to by the Pointer parameter to the number of bytes specified by the Size parameter and returns a new pointer to the block. The pointer specified by the Pointer parameter must have been created with the malloc, calloc, or realloc subroutines and not been deallocated with the free or realloc subroutines. Undefined results occur if the Pointer parameter is not a valid pointer.

What is function overloading and operator overloading?

Function overloading: C++ enables several functions of the same name to be defined, as long as these functions have different sets of parameters (at least as far as their types are concerned). This capability is called function overloading. When an overloaded function is called, the C++ compiler selects the proper function by examining the number, types and order of the arguments in the call. Function overloading is commonly used to create several functions of the same name that perform similar tasks but on different data types.

Operator overloading allows existing C++ operators to be redefined so that they work on objects of user-defined classes. Overloaded operators are syntactic sugar for equivalent function calls. They form a pleasant facade that doesn't add anything fundamental to the language (but they can improve understandability and reduce maintenance costs).

What is the difference between declaration and definition?

The declaration tells the compiler that at some later point we plan to present the definition of this declaration.
E.g.: void stars () //function declaration
The definition contains the actual implementation.
E.g.: void stars () // declarator

```
{
for(int j=10; j > =0; j--) //function body
cout << *;
cout <<>
```

What are the advantages of inheritance?

It permits code reusability. Reusability saves time in program development. It encourages the reuse of proven and debugged high-quality software, thus reducing problem after a system becomes functional.

How do you write a function that can reverse a linked-list?

```
void reverselist(void)
{
if(head==0)
return;
if(head->next==0)
return;
if(head->next==tail)
{
head->next = 0;
tail->next = head;
}
else
{
node* pre = head;
node* cur = head->next;
node* curnext = cur->next;
head->next = 0;
cur-> next = head;

for(; curnext!=0; )
{
cur->next = pre;
pre = cur;
cur = curnext;
curnext = curnext->next;
}
curnext->next = cur;
}
}
```

What do you mean by inline function?

The idea behind inline functions is to insert the code of a called function at the point where the function is called. If done carefully, this can improve the application's performance in exchange for increased compile time and possibly (but not always) an increase in the size of the generated binary executables.

Write a program that ask for user input from 5 to 9 then calculate the average

```
#include "iostream.h"
int main() {
int MAX = 4;
int total = 0;
int average;
int numb;
for (int i=0; icout << "Please enter your input between 5 and 9: ";
cin >> numb;
while ( numb<5>9) {
cout << "Invalid input, please re-enter: ";
cin >> numb;
}
total = total + numb;
}
average = total/MAX;
cout << "The average number is: " <<>return 0;
}
```

What is public, protected, private?

Public, protected and private are three access specifiers in C++.
Public data members and member functions are accessible outside the class.
Protected data members and member functions are only available to derived classes.
Private data members and member functions can��t be accessed outside the class. However there is an exception can be using friend classes.

What is scope & storage allocation of global and extern variables? Explain with an example

Extern variables: belong to the External storage class and are stored in the main memory. extern is used when we have to refer a function or variable that is implemented in other file in the same project.
The scope of the extern variables is Global.

Global variables: are variables which are declared above the main( ) function. These variables are accessible throughout the program. They can be accessed by all the functions in the program. Their default value is zero.

What is scope & storage allocation of static, local and register variables? Explain with an example.

Register variables: belong to the register storage class and are stored in the CPU registers. The scope of the register variables is local to the block in which the variables are defined. The variables which are used for more number of times in a program are declared as register variables for faster access.
Example: loop counter variables.
register int y=6;
Static variables: Memory is allocated at the beginning of the program execution and it is reallocated only after the program terminates. The scope of the static variables is local to the block in which the variables are defined.
Example:

```
#include
void decrement(){
static int a=5;
a--;
printf("Value of a:%d\n", a);
}
```

```
int main(){
decrement();
return 0;
}
```

Local variables: are variables which are declared within any function or a block. They can be accessed only by function or block in which they are declared. Their default value is a garbage value.

What are the advantages of using unions?

Union is a collection of data items of different data types.
It can hold data of only one member at a time though it has members of different data types.

If a union has two members of different data types, they are allocated the same memory. The memory allocated is equal to maximum size of the members. The data is interpreted in bytes depending on which member is being accessed.

Example:

```
union pen {
char name;
float point;
};
```

Here name and point are union members. Out of these two variables, □□point□□ is larger variable which is of float data type and it would need 4 bytes of memory. Therefore 4 bytes space is allocated for both the variables. Both the variables have the same memory location. They are accessed according to their type. Union is efficient when members of it are not required to be accessed at the same time.

Tell how to check whether a linked list is circular.

Create two pointers, each set to the start of the list. Update each as follows:

```
while (pointer1) {
pointer1 = pointer1->next;
pointer2 = pointer2->next; if (pointer2) pointer2=pointer2->next;
if (pointer1 == pointer2) {
print (\"circular\n\");
}
}
```

If a list is circular, at some point pointer2 will wrap around and be either at the item just before pointer1, or the item before that. Either way, it’s either 1 or 2 jumps until they meet.

What is virtual constructors/destructors?

Virtual destructors: If an object (with a non-virtual destructor) is destroyed explicitly by applying the delete operator to a base-class pointer to the object, the base-class destructor function (matching the pointer type) is called on the object.
There is a simple solution to this problem □□ declare a virtual base-class destructor. This makes all derived-class

destructors virtual even though they don��t have the same name as the base-class destructor. Now, if the object in the hierarchy is destroyed explicitly by applying the delete operator to a base-class pointer to a derived-class object, the destructor for the appropriate class is called.

Virtual constructor: Constructors cannot be virtual. Declaring a constructor as a virtual function is a syntax error.

What are the advantages of inheritance?

�¢ It permits code reusability.
�¢ Reusability saves time in program development.
�¢ It encourages the reuse of proven and debugged high-quality software, thus reducing problem after a system becomes functional.

Does c++ support multilevel and multiple inheritance?
Yes.

What is the difference between an ARRAY and a LIST?

Array is collection of homogeneous elements.
List is collection of heterogeneous elements.

For Array memory allocated is static and continuous.
For List memory allocated is dynamic and Random.

Array: User need not have to keep in track of next memory allocation.
List: User has to keep in Track of next location where memory is allocated.

Array uses direct access of stored members, list uses sequencial access for members.

What is a template?

Templates allow to create generic functions that admit any data type as parameters and return value without having to overload the function with all the possible data types. Until certain point they fulfill the functionality of a macro. Its prototype is any of the two following ones:

What is the difference between class and structure?

Structure: Initially (in C) a structure was used to bundle different type of data types together to perform a particular functionality. But C++ extended the structure to contain functions also. The major difference is that all declarations inside a structure are by default public.
Class: Class is a successor of Structure. By default all the members inside the class are private.

What is encapsulation?

Packaging an objects variable within its methods is called encapsulation.

What is a COPY CONSTRUCTOR and when is it called?

A copy constructor is a method that accepts an object of the same class and copies it’s data members to the object on the left part of assignment:

```
class Point2D{
int x; int y;
public int color;
```

```
protected bool pinned;
public Point2D() : x(0) , y(0) {}//default (no argument) constructor
public Point2D( const Point2D & ) ;
};
Point2D::Point2D( const Point2D & p )
{
this->x = p.x;
this->y = p.y;
this->color = p.color;
this->pinned = p.pinned;
}
main(){yu
Point2D MyPoint;
MyPoint.color = 345;
Point2D AnotherPoint = Point2D( MyPoint ); // now AnotherPoint has color = 345
```

Program: To calculate the factorial value using recursion.

Program: To calculate the factorial value using recursion.

```
#include
int fact(int n);

int main() {
int x, i;
printf("Enter a value for x: \n");
scanf("%d", &x);
i = fact(x);
printf("\nFactorial of %d is %d", x, i);
return 0;
}

int fact(int n) {
/* n=0 indicates a terminating condition */
if (n <= 0) {
return (1);
}else {
/* function calling itself */
return (n * fact(n - 1));
/*n*fact(n-1) is a recursive expression */
}
}
```

Output:

```
Enter a value for x:4

Factorial of 4 is 24
```

swap 2 numbers without using third variable?

```
#include
void main()
{
int a,b;
printf("enter number1: ie a");
scanf("%d",a);
printf("enter number2:ie b ");
scanf("%d",b);
printf(value of a and b before swapping is a=%d,b=%d"a,b);
a=a+b;
b=a-b;
a=a-b;
printf(value of a and b after swapping is a=%d,b=%d"a,b);
}
```

Write a C++ Program to check whether a number is prime number or not?

```
#include
#include

void main()
{
clrscr();
int n,i,flag=1;
cout<<"Enter any number:";
cin>>n;


for(i=2;i<=n/2;++i)
{
if(n%i==0)
{
flag=0;
break;
}
}
if(flag)
cout<<"\n"<<n<<" is="" a="" prime="" number";
else
cout<<"\n"<<n<<" is="" not="" a="" prime="" number";
getch();
}</n<<"></n<<">
```

Write a program in c to replace any vowel in a string with z?

```
// Replace all vowels in str with 'z'
void replaceWithZ(char* str) {
int i = 0;
while(str[i] != 'z') {
if(isVowel(str[i])) {
```

```
str[i] = 'z';
}
++i;
}
}
// Returns 1 if ch is a vowel, 0 otherwise
int isVowel(const char ch) {

switch(ch) {
case 'a':case 'A':
case 'e':case 'E':
case 'i':case 'I':
case 'o':case 'O':
case 'u':case 'U':
return 1;
}
return 0;
}
// Sample call
int main() {
char str[] = "HELLO";
printf("%s\n", str);
replaceWithZ(str);
printf("%s\n", str);
return 0;
}
```

Question from Java

What is the final keyword denotes?

final keyword denotes that it is the final implementation for that method or variable or class. You can’t override that method/variable/class any more.

What is the significance of ListIterator?

You can iterate back and forth.

What is the major difference between LinkedList and ArrayList?

Linked List are meant for sequential accessing. ArrayList are meant for random accessing.

What is nested class?

If all the methods of a inner class is static then it is a nested class.

What is inner class?

If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.

What is composition?

Holding the reference of the other class within some other class is known as composition.

What is aggregation?

It is a special type of composition. If you expose all the methods of a composite class and route the method call to the composite method through its reference, then it is called aggregation.

What are the methods in Object?

clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString

Can you instantiate the Math class?

You can’t instantiate the math class. All the methods in this class are static. And the constructor is not public.

What is singleton? - It is one of the design pattern. This falls in the creational pattern of the design pattern. There will be only one instance for that entire JVM. You can achieve this by having the private constructor in the class. For eg., public class Singleton { private static final Singleton s = new Singleton(); private Singleton(){}public static Singleton getInstance(){return s; }// all non static methods ¦ }