

## IBS Software Services Technical Interview Questions

Q1. What is the difference between declaration and definition?

**ANS:**

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 <<>`

Q2. What do you mean by inline function?

**ANS:**

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 applications performance in exchange for increased compile time and possibly (but not always) an increase in the size of the generated binary executables.

Q3. What is the final keyword denotes?

**ANS:**

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

Q4. What is composition?

**ANS:**

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

Q5. What are the methods in Object?

**ANS:**

Clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString.

Q6. What are the advantages of inheritance?

**ANS:**

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.

Q7. What is public, protected, private?

**ANS:**

1. Public, protected and private are three access specifiers in C++.
2. Public data members and member functions are accessible outside the class.
3. Protected data members and member functions are only available to derived classes.
4. Private data members and member functions can not be accessed outside the class. However there is an exception can be using friend classes.

Q8. What is Boyce Codd Normal form?

**ANS:**

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  $a \rightarrow b$ , where a and b is a subset of R, at least one of the following holds:

- \*  $a \rightarrow b$  is a trivial functional dependency (b is a subset of a)
- \* a is a super key for schema R

Q9. What is the difference between realloc() and free()?

**ANS:**

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.

Q10. What is singleton?

**ANS:**

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}
```

Q11. What is the major difference between Linked List and Array List?

**ANS:**

Linked List is meant for sequential accessing. Array List is meant for random accessing.

Q12. Can you instantiate the Math class?

**ANS:**

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

Q13. What is inner class?

**ANS:**

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

Q14. What do you mean by inheritance?

**ANS:**

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.

Q15. How do you find out if a linked-list has an end? (i.e. the list is not a cycle)

**ANS:**

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.

Q16. What is aggregation?

**ANS:**

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.

Q17. What is skeleton and stub? What is the purpose of those?

**ANS:**

Stub is a client side representation of the server, which takes care of communicating with the remote server. Skeleton is the server side representation. But that is no more in use... it is deprecated long before in JDK.

Q18. What is an object?

**ANS:**

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

Q19. What is virtual class and friend class?

**ANS:**

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

Q20. What is function overloading?

**ANS:**

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.

Q21. What is the significance of List Iterator?

**ANS:**

You can iterate back and forth.

Q22. What is nested class?

**ANS:**

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

Q23. Write a program that ask for user input from 5 to 9 then calculate the average

**ANS:**

```
#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;
```

```
}
```

Q24. How do you write a function that can reverse a linked-list?

**ANS:**

```
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;
}
}
```

Q25. What is operator overloading?

**ANS:**

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).