OUR EDUCATION

Success Comes in Way...

Interview Question Answers on C++

1. What is C++?

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 consoleonly 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++.

2. 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.

3. 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.

4. 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
Customer is king inc.,

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).

5. 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 <<>
```

6. 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.

7. 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;
Customer is king inc.,
}
curnext->next = cur;
}
}
```

8. 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.

9. 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;i
{
cout << "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: " << average << "\n";
return 0;
}
```

10. Write a short code using C++ to print out all odd number from 1 to 100 using a for loop

```
for( unsigned int i = 1; i < = 100; i++ )
if( i & 0x00000001 )
cout << i << \",\";
```

11. 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.

12. Write a function that swaps the values of two integers, using int* as the argument type.

```
void swap(int* a, int*b) { int t; t = *a; *a = *b; *b = t; }
```

13. 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\");
}
}
```

OK, why does this work?
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.

14. 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.

15. 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.

16. 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:

template function_declaration; template function_declaration;

17. What is the difference between an object and a class?

I. Classes and objects are separate but related concepts.
Every object belongs to a class and every class contains one or more related objects.

II. A Class is static. All of the attributes of a class are fixed before, during, and after the execution of a program. The attributes of a class don't change.
The class to which an object belongs is also (usually) static. If a particular object belongs to a certain class at the time that it is created then it almost certainly will still belong to that class right up until the time that it is destroyed.

III. An Object on the other hand has a limited lifespan. Objects are created and eventually destroyed. Also during that lifetime, the attributes of the object may undergo significant change.

18. What is a class?

Class is a user-defined data type in C++. It can be created to solve a particular kind of problem. After creation the user need not know the specifics of the working of a class.

19. What is friend function?

As the name suggests, the function acts as a friend to a class. As a friend of a class, it can access its private and protected members. A friend function is not a member of the class. But it must be listed in the class definition.

20. Which recursive sorting technique always makes recursive calls to sort subarrays that are about half size of the original array?

Mergesort always makes recursive calls to sort subarrays that are about half size of the original array, resulting in O(n log n) time.

21. What are virtual functions?

A virtual function allows derived classes to replace the implementation provided by the base class. The compiler makes sure the replacement is always called whenever the object in question is actually of the derived class, even if the object is accessed by a base pointer rather than a derived pointer. This allows algorithms in the base class to be replaced in the derived class, even if users don't know about the derived class.

22. What is the difference between an external iterator and an internal iterator? Describe an advantage of an external iterator.

An internal iterator is implemented with member functions of the class that has items to step through. .An external iterator is implemented as a separate class that can be "attach" to the object that has items to step through. .An external iterator has the advantage that many difference iterators can be active simultaneously on the same object.

23. What is a scope resolution operator?

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

24. What do you mean by pure virtual functions?

A pure virtual member function is a member function that the base class forces derived classes to provide. Normally these member functions have no implementation. Pure virtual functions are equated to zero.
class Shape { public: virtual void draw() = 0; };

25. What is polymorphism? Explain with an example?

"Poly" means "many" and "morph" means "form". Polymorphism is the ability of an object (or reference) to assume (be replaced by) or become many different forms of object.

Example: function overloading, function overriding, virtual functions. Another example can be a plus ‘+’ sign, used for adding two integers or for using it to concatenate two strings.

26. Why are arrays usually processed with for loop?

The real power of arrays comes from their facility of using an index variable to traverse the array, accessing each element with the same expression a[i]. All the is needed to make this work is a iterated statement in which the variable i serves as a counter, incrementing from 0 to a.length -1. That is exactly what a loop does.

27. What is an HTML tag?

An HTML tag is a syntactical construct in the HTML language that abbreviates specific instructions to be executed when the HTML script is loaded into a Web browser. It is like a method in Java, a function in C++, a procedure in Pascal, or a subroutine in FORTRAN.

28. How do you decide which integer type to use?

It depends on our requirement. When we are required an integer to be stored in 1 byte (means less than or equal to 255) we use short int, for 2 bytes we use int, for 8 bytes we use long int.

A char is for 1-byte integers, a short is for 2-byte integers, an int is generally a 2-byte or 4-byte integer (though not necessarily), a long is a 4-byte integer, and a long long is a 8-byte integer.

29. What’s the best way to declare and define global variables?

The best way to declare global variables is to declare them after including all the files so that it can be used in all the functions.

30. What does extern mean in a function declaration?

Using extern in a function declaration we can make a function such that it can used

outside the file in which it is defined.
An extern variable, function definition, or declaration also makes the described variable or function usable by the succeeding part of the current source file. This declaration does not replace the definition. The declaration is used to describe the variable that is externally defined.
If a declaration for an identifier already exists at file scope, any extern declaration of the same identifier found within a block refers to that same object. If no other declaration for the identifier exists at file scope, the identifier has external linkage.

31. What is the difference between char a[] = “string”; and char *p = “string”;?

In the first case 6 bytes are allocated to the variable a which is fixed, where as in the second case if *p is assigned to some other value the allocate memory can change.

32. What does extern mean in a function declaration?

It tells the compiler that a variable or a function exists, even if the compiler hasn’t yet seen it in the file currently being compiled. This variable or function may be defined in another file or further down in the current file.

33. Explain the scope resolution operator.

It permits a program to reference an identifier in the global scope that has been hidden by another identifier with the same name in the local scope.

34. What is a default constructor?

Default constructor WITH arguments class B { public: B (int m = 0) : n (m) {} int n; }; int main(int argc, char *argv[]) { B b; return 0; }

35. What is the difference between a copy constructor and an overloaded assignment operator?

A copy constructor constructs a new object by using the content of the argument object. An overloaded assignment operator assigns the contents of an existing object to another existing object of the same class.

36. When should you use multiple inheritance?

There are three acceptable answers: "Never," "Rarely," and "When the problem domain cannot be accurately modeled any other way."

37. Explain the ISA and HASA class relationships. How would you implement each in a class design?

A specialized class "is" a specialization of another class and, therefore, has the ISA relationship with the other class. An Employee ISA Person. This relationship is best implemented with inheritance. Employee is derived from Person. A class may have an instance of another class. For example, an employee "has" a salary, therefore the Employee class has the HASA relationship with the Salary class. This relationship is best implemented by embedding an object of the Salary class in the Employee class.

38. When is a template a better solution than a base class?

When you are designing a generic class to contain or otherwise manage objects of other types, when the format and behavior of those other types are unimportant to their containment or management, and particularly when those other types are unknown (thus, the genericity) to the designer of the container or manager class.

39. What is an explicit constructor?

A conversion constructor declared with the explicit keyword. The compiler does not use an explicit constructor to implement an implied conversion of types. It’s purpose is reserved explicitly for construction.

40. What is the Standard Template Library (STL)?

A library of container templates approved by the ANSI committee for inclusion in the standard C++ specification.
A programmer who then launches into a discussion of the generic programming model, iterators, allocators, algorithms, and such, has a higher than average understanding of the new technology that STL brings to C++ programming.

41. In C++, what is the difference between method overloading and method overriding?

Overloading a method (or function) in C++ is the ability for functions of the same name to be defined as long as these methods have different signatures (different set of parameters). Method overriding is the ability of the inherited class rewriting the virtual method of the base class.

42. What are the defining traits of an object-oriented language?

The defining traits of an object-oriented langauge are:
* encapsulation
* inheritance
* polymorphism

43. What is class invariant?

A class invariant is a condition that defines all valid states for an object. It is a logical condition to ensure the correct working of a class. Class invariants must hold when an object is created, and they must be preserved under all operations of the class. In particular all class invariants are both preconditions and post-conditions for all operations or member functions of the class.

44. What is a protocol class?

An abstract class is a protocol class if:

- it neither contains nor inherits from classes that contain member data, non-virtual functions, or private (or protected) members of any kind.
- it has a non-inline virtual destructor defined with an empty implementation,
- all member functions other than the destructor including inherited functions, are declared pure virtual functions and left undefined.

45. What is a parameterized type?

A template is a parameterized construct or type containing generic code that can use or manipulate any type. It is called parameterized because an actual type is a parameter of the code body. Polymorphism may be achieved through parameterized types. This type of polymorphism is called parametric polymorphism. Parameteric polymorphism is the mechanism by which the same code is used on different types passed as parameters.

46. What is reflexive association?

The 'is-a' is called a reflexive association because the reflexive association permits classes to bear the is-a association not only with their super-classes but also with themselves. It differs from a 'specializes-from' as 'specializes-from' is usually used to describe the association between a super-class and a sub-class.

For example: **Printer is-a printer.**

47. What are proxy objects?

Objects that points to other objects are called proxy objects or surrogates. Its an object that provides the same interface as its server object but does not have any functionality. During a method invocation, it routes data to the true server object and sends back the return value to the object.

48. Describe the main characteristics of static functions.

The main characteristics of static functions include,

- It is without the a this pointer,
- It can't directly access the non-static members of its class
- It can't be declared const, volatile or virtual.
- It doesn't need to be invoked through an object of its class, although for convenience, it may.

49. How can a '::' operator be used as unary operator?

The scope operator can be used to refer to members of the global namespace. Because the global namespace doesn’t have a name, the notation :: member-name refers to a member of the global namespace. This can be useful for referring to members of global namespace whose names have been hidden by names declared in nested local scope. Unless we specify to the compiler in which namespace to search for a declaration, the compiler simple searches the current scope, and any scopes in which the current scope is nested, to find the declaration for the name.

50. What is a concrete class?

A concrete class is used to define a useful object that can be instantiated as an automatic variable on the program stack. The implementation of a concrete class is defined. The concrete class is not intended to be a base class and no attempt to minimize dependency on other classes in the implementation or behavior of the class.