

C++ Interview Questions

1. **What is C++?**

C++ is created by Bjarne Stroustrup of AT&T Bell Labs as an extension of C, C++ is an object-oriented computer language used in the development of enterprise and commercial applications. Microsoft's Visual C++ became the premier language of choice among developers and programmers.

2. **What are the basic concepts of object oriented programming?**

It is necessary to understand some of the concepts used extensively in object oriented programming. These include

Objects

Classes

Data abstraction and encapsulation

Inheritance

Polymorphism

Dynamic Binding

Message passing

3. **Define inheritance?**

The mechanism of deriving a new class (derived) from an old class (base class) is called inheritance. It allows the extension and reuse of existing code without having to rewrite the code from scratch. Inheritance is the process by which objects of one class acquire properties of objects of another class.

4. **Define polymorphism?**

Polymorphism means one name, multiple forms. It allows us to have more than one function with the same name in a program. It allows us to have overloading of operators so that an operation can exhibit different behaviours in different instances.

5. **What are the features of C++ different from C?**

All the features of C are similar to C++ except some features, such as polymorphism, operator overloading which are supported in C++ but not in C language.

Both C and C++ language is similar in their functionality but C++ provides

with more tools and options.

6. What is encapsulation?

The wrapping up of data and functions into a single unit (called class) is known as encapsulation. Encapsulation containing and hiding information about an object, such as internal data structures and code.

7. What is message passing?

An object oriented program consists of a set of objects that communicate with each other. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

8. What are tokens in C++?

The smallest individual units of a program is known as tokens. c++ has the following tokens :

Keywords

Identifiers

Constants

Strings

Operators

9. What is the use of enumerated data type?

An enumerated data type is another user defined type which provides a way for attaching names to numbers thereby increasing comprehensibility of the code. The enum keyword automatically enumerates a list of words by assigning them values 0,1,2, and so on.

10. What is the use of default constructor?

A constructors that accepts no parameters is called the default constructor. If no user-defined constructor exists for a class A and one is needed, the compiler implicitly declares a default parameterless constructor A::A(). This constructor is an inline public member of its class. The compiler will implicitly define A::A() when the compiler uses this constructor to create an object of type A. The constructor will have no constructor initializer and a null body.

11. Define Constructors?

A constructor is a member function with the same name as its class. The constructor is invoked whenever an object of its associated class is created. It is called constructor because it constructs the values of data members of the class.

12. How variable declaration in c++ differs that in c?

C requires all the variables to be declared at the beginning of a scope but in c++ we can declare variables anywhere in the scope. This makes the programmer easier to understand because the variables are declared in the context of their use.

13. Define destructors?

A destructor is called for a class object when that object passes out of scope or is explicitly deleted.

A destructors as the name implies is used to destroy the objects that have been created by a constructors.

Like a constructor , the destructor is a member function whose name is the same as the class name but is preceded by a tilde.

14. What is a class?

A class is a collection of objects.

15. what is the difference between c & c++?

- C++ is an object oriented programming but C is a procedure oriented programming.
- C is super set of C++.
- C can't support inheritance, function overloading, method overloading etc. but c++ can do this.
- In c program the main function could not return a value but in the c++ the main function should return a value.

16. What are the few advantages of Inline function?

It offers an improved macro facility.

By using the inline functions, the user can split a large function with many nested modules of statement blocks into many small inline functions.

17. What is copy constructor?

Copy constructor is a constructor function with the same name as the class and used to make deep copy of objects.

18. What is default constructor?

A default constructor is a constructor that either has no parameters, or if it

has parameters, all the parameters have default values.

19. What is a scope resolution operator?

The scope resolution operator 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.

20. What is the difference between Object and Instance?

An instance of a user-defined type is called an object. We can instantiate many objects from one class.

An object is an instance of a class.

21. What is the difference between macro and inline?

Inline follows strict parameter type checking, macros do not.

Macros are always expanded by preprocessor, whereas compiler may or may not replace the inline definitions.

22. How variable declaration in c++ differs that in c?

C requires all the variables to be declared at the beginning of a scope but in c++ we can declare variables anywhere in the scope. This makes the programmer easier to understand because the variables are declared in the context of their use.

23. What is multiple inheritance?

A class can inherit properties from more than one class which is known as multiple inheritance.

24. what is the use of virtual destructor in c++?

A destructor is automatically called when the object is destroyed.

A virtual destructor in C++ is used primarily to prevent resource leaks by performing a clean-up of the object.

25. What do you mean by reference variable in c++?

A reference variable provides an alias to a previously defined variable.
Data -type & reference-name = variable name

26. What is iterator class?

Iterator class provides an access to the class which are inside the containers(it holds a group of objects in an organized way).

The containers include the data structure, class and abstract data type.

27. What are the types of declarations in C++?

There are so many types of declaration in C++ are :

Variable declaration

Constant declaration

Function declaration

Object declaration

28. What are Smart pointers?

Smart pointers are almost similar to pointers with additional features such as automatic destruction of a variable when it becomes out of scope and the throwing of exceptions that ensures the proper destruction of the dynamically allocated objects.

29. Explain function template?

Function template provides a means to write generic functions for different data types such as integer, long, float or user defined objects.

30. Explain class template?

Class template provides a means to write a generic class for different types so that a class can have members based on generic types that do not need to be defined at the moment of creating the class or whose members use these generic types.

31. What is difference between function overloading and operator overloading?

A function is overloaded when same name is given to different function.

While overloading a function, the return type of the functions need to be the same.

32. What are the advantages of inheritance?

Code reusability

Saves time in program development.

33. What is a dynamic constructor?

The constructor can also be used to allocate memory while creating objects.

Allocation of memory to objects at the time of their construction is known as dynamic construction of objects.

The memory is allocated with the help of the new operator.

34. What is the difference between an Array and a List?

The main difference between an array and a list is how they internally store the data. whereas Array is collection of homogeneous elements. List is collection of heterogeneous elements.

35. What is the use of 'using' declaration?

A using declaration makes it possible to use a name from a namespace.

36. What is the difference between a template class and class template?

Template classA generic definition or a parameterized class not instantiated until the client provides the needed information. It's jargon for plain templates.

Class templateA class template specifies how individual classes can be constructed much like the way a class specifies how individual objects can be constructed. It's jargon for plain classes.

37. What is friend function?

The function declaration should be preceded by the keyword friend.

The function definitions does not use either the keyword or the scope operator ::

The functions that are declared with the keyword friend as friend function.

Thus, a friend function is an ordinary function or a member of another class.

38. What is a scope resolution operator?

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

39. 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. Any class containing any pure virtual function cannot be used to create object of its own type.

40. What is a conversion constructor?

A converting constructor is a single-parameter constructor that is declared without the function specifier explicit.

The compiler uses converting constructors to convert objects from the type of the first parameter to the type of the converting constructor's class.

41. What is a container class? What are the types of container classes?

A container class is a class that is used to hold objects in memory or external storage.

A container class acts as a generic holder.

A container class has a predefined behavior and a wellknown interface.

A container class is a supporting class whose purpose is to hide the topology used for maintaining the list of objects in memory.

When a container class contains a group of mixed objects, the container is called a heterogeneous container; when the container is holding a group of objects that are all the same, the container is called a homogeneous container.

42. What is Associative container?

Associative containers are designed to support direct access to elements using keys. They are not sequential. There are four types of associative containers :

Set

Multiset

Map

Multimap

43. What is an iterator?

Iterators are like pointers. They are used to access the elements of containers thus providing a link between algorithms and containers. Iterators are defined for specific containers and used as arguments to algorithms.

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

The defining traits of an object-oriented language are :

Encapsulation

Inheritance

Polymorphism

45. What is this pointer?

It is a pointer that points to the current object. This can be used to access the members of the current object with the help of the arrow operator.

46. Name some pure object oriented languages?

Smalltalk

Java

Eiffel

Sather

47. What is encapsulation?

Encapsulation (or information hiding) is the process of combining data and functions into a single unit called class.

48. What is problem with Runtime type identification?

The run time type identification comes at a cost of performance penalty. Compiler maintains the class.

49. What are the differences between new and malloc?

New initializes the allocated memory by calling the constructor. Memory allocated with new should be released with delete.

Malloc allocates uninitialized memory.

The allocated memory has to be released with free. new automatically calls the constructor while malloc(doesn't)

50. What is conversion operator?

You can define a member function of a class, called a conversion function, that converts from the type of its class to another specified type.

51. What do you mean by implicit conversion?

Whenever data types are mixed in an expression then c++ performs the conversion automatically.

Here smaller type is converted to wider type.

Example : in case of integer and float integer is converted into float type.

52. What are virtual functions?

The virtual functions must be members of some class.

They cannot be static members.

They are accessed by using object pointers.

A virtual function can be a friend of another class.

53. What is the main purpose of overloading operators?

The main purpose of operator overloading is to minimize the chances of occurrence of errors in a class that is using the overload operators.

It also helps in redefining the functionalities of the operators to improve their performance.

Operator overloading also makes the program clearer, readable and more understandable by using common operators, such as +, =, and [].

54. What is a friend?

Friends can be either functions or other classes. The class grants friends unlimited access privileges.

55. What is stack unwinding?

Stack unwinding is a process in which a destructor is invoked in a particular program for destroying all the local objects in the stack between throwing and catching of an exception.

56. What is the difference between class and structure?

By default, the members of structures are public while that for class is private. Structures don't provide something like data hiding which is provided by the classes.

Structures contain only data while class binds both data and member functions.

57. What are storage qualifiers in C++ ?

Const keyword indicates that memory once initialized, should not be altered by a program.

Volatile keyword indicates that the value in the memory location can be altered even though nothing in the program.

Mutable keyword indicates that particular member of a structure or class can be altered even if a particular structure variable, class, or class member function is constant.

58. What is virtual class and friend class?

Friend classes are used when two or more classes and virtual base class aids

in multiple inheritance.

Virtual class is used for run time polymorphism when object is linked to procedure call at run time.

59. What is an abstract base class?

An abstract class is a class that is designed to be specifically used as a base class. An abstract class contains at least one pure virtual function.

60. What is dynamic binding?

Dynamic binding (also known as late binding) means that the code associated with a given procedure call is not known until the time of the call at run time. It is associated with polymorphism and inheritance.

61. What are the benefits of object oriented programming(OOP)?

Software reusability

Code sharing

Rapid prototyping

Information hiding

.

62. What is the form of assignment statement?

Variable = expression (or constant)

63. What is the main purpose of overloading operators?

The main purpose of operator overloading is to minimize the chances of occurrence of errors in a class that is using the overloaded operators.

64. What is this pointer?

When a member function is invoked, the invoking object's pointer is passed implicitly as an argument. This pointer is called this pointer.

65. What is scope resolution operator?

The Scope resolution operator(::) can be used to define the member functions of a program outside the boundary of a class and not within the class specifier.

66. What are static members and static functions?

Static members are
Created and initialized only once.
Shared among all the class objects.

Static functions are
Similar to the static variables and are associated with the class.
Can only access static variables of a class.
Can also be called using the scope resolution operator.

67. What are the components of a class?

A class consists of two components,
Data members
Methods

68. What is the advantage of using templates?

Templates provide a means to write generic functions and classes for different data types.

Templates are sometimes called parameterized types.

Templates can significantly reduce source code size and increase code flexibility without reducing type safety.

69. Can a function overloading depend only on passing by value and passing by reference?

No, the reason is that whether a function is called the passing a parameter as a value or by reference, it appears similar to the caller of the function.

70. Is it possible to use a new for the reallocation of pointers?

The reallocation of pointers cannot be done by using new. It can be done by using the realloc() operator.

71. What are the types of storage qualifiers in C++?

C++ includes three storage qualifiers:

Const : A const variable is one that the program may not modify except through initialization when the variable is declared.

Volatile: A volatile type qualifier tells the compiler that the program could change the variable.

Mutable: A const member function may modify a data member only if the data member is declared with the mutable qualifier.

72. What are the advantages of using on iterator?

Iterator interfaces(API) are the same for all the containers. For example, a container list can internally have doubly linked list or singly list, but its corresponding iterator interface that is used to access its elements is always the same.

(iter->next)

73. What are data members?

Data members are variables of any type(in-built or user defined).

74. What are the types of statements in c++?

A program in any language basically consists of statements. Statements symbolize instructions. There are many categories of statements.

Expression statement

Assignment statement

Selection statement

Iteration statement

Jump statement

75. What is initialization?

Initialization is a process of assigning a value to a variable at the time of declaration.

76. What is the difference between a vector and a map?

A vector is a sequential container, i.e., all the elements are in a sequence, whereas a map is an association container, i.e., all elements are stored in the form of a key value association pair.

77. What are the advantages of using cin and cout compared to scanf(...) and printf(...), respectively?

Compared to the standard C functions printf() and scanf(), the usage of the cin and cout is more type safe.

The format strings, which are used with printf() and scanf() can define wrong format specifies for their arguments, for which the compiler does not warn.

In contrast, argument checking with c in and cout is performed by the compiler.

C in and Cout are stream classes that could be used to receive and print

objects respectively.

78. Explain copy constructor?

A copy constructor is a special type of constructor which initializes all the data members of the newly created object by copying the contents of an existing object. The compiler provides a default copy constructor.

`Class_name new _ object (existing object);`

79. What are the advantages of operator overloading?

Operator overloading is used to provide some extra features, behaviors and abilities to the users of a particular class. This feature in C++ helps in controlling the functions performed by an operator and reduces the chance of occurrence of errors in a program.

80. What is a dangling pointer?

When the location of the deallocated memory is pointed by the pointer even after the deletion or allocation of objects is done, without the modification in the value of the pointer, then this type of pointer is called a dangling pointer.

81. What are shallow and deep copies?

A shallow copy is used to copy actual values of the data. It means, if the pointer points to dynamically allocated memory, the new object's pointer in the copy still points to items inside the old objects and the returned object will be left pointing to items that are no longer in scope.

A copy of the dynamically allocated objects is created with the help of deep copy. This is done with the help of an assignment operator, which needs to be overloaded by the copy constructor.

82. How can you return the current involving object from its member function?

`return(*this);`

83. What is the difference between prefix and postfix versions of operator++()?

The prefix and postfix versions of operator ++() can be differentiated on the basis of arguments defined.

The postfix operator ++() consists of a dummy parameter of int datatype; whereas, a dummy parameter is not found in the prefix operator ++().

84. Can a static member function access member variable of an object?

No, because to access the member variable of an object inside its member function, this pointer is required. Since static functions are class functions,

this pointer will not be passed as its arguments.

85. What is the advantages of using the Inline function?

An inline keyword before a function suggests the compiler to insert the complete body of the function wherever that function is invoked.

Inline expansion is typically used to eliminate the inherent cost involved in calling a function.

It is typically used for functions that need quick execution.

86. What are all the operators that cannot be overloaded?

Direct member access operator

De-reference pointer to class member operator.*

Scope resolution operator::

Conditional operator ?:

Sizeof operator sizeof

87. Can a function be overloaded based on return types?

Function signature does not depend on the return type. So overloading cannot be resolved by the return type alone.

88. What do you mean by a public member?

A member declared as public is a public member.

It can be accessed freely in a program.

89. Is recursion allowed in inline functions?

The recursion is allowed in inline function but practically, the inline functions and their properties do not remain inside the program. Moreover, the compiler is not sure about the depth of the recursion at the time of compilation.

90. What is virtual function?

A virtual function is a member function that is declared within a base class and redefined by a derived class .To create a virtual function, the function declaration in the base class is preceded by the keyword virtual.

91. How can a struct in C++ differs from a struct in C?

The differences between struct in C++ and C are listed in the following points:

In C and C++, the variables of the structures are public; however, in C, the variable cannot be declared as private or protected. On the contrary, in C++, the variables can be declared as private or protected.

In C, the concept of inheritance is not supported. In C++, the concept of inheritance is fully supported.

On declaring a struct in C, the addition of the struct keyword is must. On the contrary, there is no need of the struct keyword on declaring struct in C++.

In C, the initialization cannot be done outside the scope of a structure. However, in C++, the initialization can be done outside the scope of a structure.

In C, structures do not have direct functions or methods.

92. How the keyword struct is different from the keyword class in C++?

In C++, a class is similar to a struct with the exception that, by default, all the members of a class are private; while the members of a struct are public. Encapsulation is not supported by structures but supported by classes.

93. Define pure virtual function?

Pure virtual function is defined as a virtual function in a base class. It is implemented in a derived class. A program may not declare an instance of a class that has a pure virtual function.

94. Define a conversion constructor?

A conversion constructor is a single argument constructor. It is used by the compiler to convert a type of objects as an argument to a class type.

95. What is a default constructor?

A zero argument constructor or a constructor in which all the arguments have default values is called a default constructor.

96. What is difference between template and macro?

A template can be used to create a family of classes or function. A template describes a set of related classes or set of related functions in which a list of parameters in the declaration describe how the members of the set vary. Identifiers that represent statements or expressions are called macros.

97. What is reference?

Reference is a name that acts as an alias, or alternative name, for a previously defined variable or an object.

98. What are the access specifier in c++?

There are three types of access specifier in c++ . They are

Public
protected
private

99. What is difference between C++ and Java?

C++ has pointers Java does not.

Java is the platform independent as it works on any type of operating systems.

java has no pointers where c ++ has pointers.

Java has garbage collection C++ does not.

100. What is namespace?

The C++ language provides a single global namespace.Namespaces allow to group entities like classes, objects and functions under a name.

101. 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.Explicit constructors are simply constructors that cannot take part in an implicit conversion.

102.What is the use of storage class specifiers?

A storage class specifier is used to refine the declaration of a variable, a function, and parameters. The following are storage class specifiers :

auto
register
static
extern

103.what is assignment operator in c++?

Default assignment operator handles assigning one object to another of the same class. Member to member copy (shallow copy).

104.Can destructor be private?

Yes destructors can be private. But according it is not advisable to have destructors to be private.

105. What is stringstream?

stringstream provides an interface to manipulate strings as if they were input/output streams.

‹ stringstream › to define several classes that support iostreams operations on sequences stored in an allocated array of char object.

106 What are the types of STL containers?

deque

hash map

hashmultimap

hash_multiset

hashset

list

map

multimap

multiset

set

vector

.

107. 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.

108. What do you mean by inline function?

An inline function is a function that is expanded inline when invoked. ie. the compiler replaces the function call with the corresponding function code. An inline function is a function that is expanded in line when it is invoked. That is the compiler replaces the function call with the corresponding function code (similar to macro).

109. What is a template?

A template can be used to create a family of classes or function. A template describes a set of related classes or set of related functions in which a list of

parameters in the declaration describe how the members of the set vary.

110. 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 its members to the object on the left part of assignment.

111. 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.

112. What is a virtual destructor?

The simple answer is that a virtual destructor is one that is declared with the virtual attribute.

113. 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.

114. What is STL? and what are the components of stl?

A collection of generic classes and functions is called as Standard Template Library (STL). The stl components are

containers

Algorithm

Iterators

.

115. What is a modifier?

A modifier, also called a modifying function is a member function that changes the value of at least one data member.

In other words, an operation that modifies the state of an object.

Modifiers are also known as mutators.

116. What is an adaptor class or Wrapper class?

A class that has no functionality of its own. Its member functions hide the use of a third party software component or an object with the non-compatible interface or a non-objectoriented implementation.

117. 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.

118. 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.

119. What is the difference between the message and method?

Message : *Objects communicate by sending messages to each other. A message is sent to invoke a method.*

Method : *Provides response to a message and it is an implementation of an operation.*

120. Is it possible to use a new for the reallocation of pointers?

The reallocation of pointers cannot be done by using new. It can be done by using the realloc() operator.

121. How can we access protected and private members of a class?

In the case of members protected and private, these could not be accessed from outside the same class at which they are declared. This rule can be transgressed with the use of the friend keyword in a class, so we can allow an external function to gain access to the protected and private members of a class.

122. What do you mean by late binding?

Late binding refers to function calls that are not resolved until run time.

Virtual functions are used to achieve late binding.

When access is via a base pointer or reference, the virtual function actually called is determined by the type of object pointed to by the pointer.

123. What do you mean by early binding?

Early binding refers to the events that occur at compile time.

Early binding occurs when all information needed to call a function is known at compile time.

Examples of early binding include normal function calls, overloaded function calls, and overloaded operators.

The advantage of early binding is efficiency.