

C++ LANGUAGE INTERVIEW QUESTIONS

1. What is C++? What are the advantages of C++?

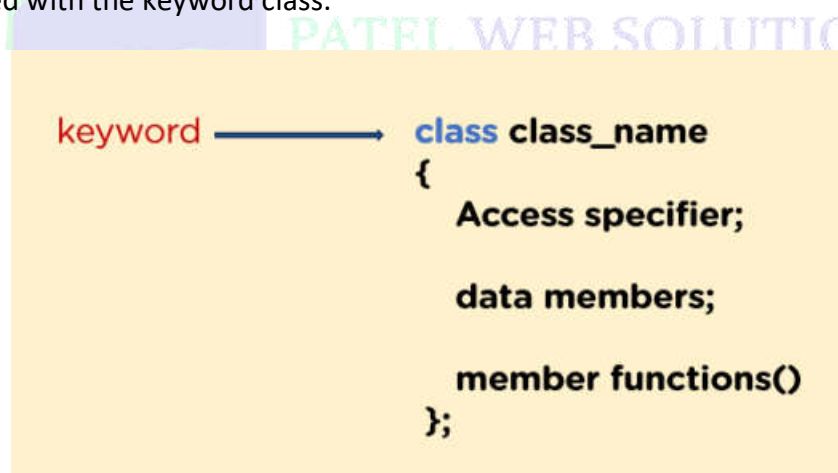
Ans. C++ is an object-oriented programming language that was introduced to overcome the jurisdictions where C was lacking. By object-oriented we mean that it works with the concept of polymorphism, inheritance, abstraction, encapsulation, object, and class.

Advantages of C++:

- C++ is an OOPs language that means the data is considered as objects.
- C++ is a multi-paradigm language; In simple terms, it means that we can program the logic, structure, and procedure of the program.
- Memory management is a key feature in C++ as it enables dynamic memory allocation.

2. What are classes and objects in C++?

Ans. A class is like a blueprint of an object. It is a user-defined data type with data members and member functions and is defined with the keyword class.



You define objects as an instance of a class. Once it creates the object, then it can operate on both data members and member functions.

3. What are access modifiers?

Ans. You use access modifiers to define accessibility for the class members. It defines how to access the members of the class outside the class scope.

There are three types of access modifiers:

- Private
- Public
- Protected

4. Difference between equal to (==) and assignment operator(=)?

Ans. The equal to operator == checks whether two values are equal or not. If equal, then it's true; otherwise, it will return false.

The assignment operator = allots the value of the right-side expression to the left operand.

5. Which among the following operators cannot be overloaded?

Ans. ?: operator cannot be overloaded because it is not syntactically possible.

6. Can you compile a program without the main function?

Ans. Yes, you can compile a program without the main function, but you cannot run or execute the program because the main() function is the entry point, from where all the execution begins. And without the entry point, then you can execute the program.

7. What is std in C++?

Ans. std is a standard namespace in C++

8. What do you understand about polymorphism in C++?

Ans. The term polymorphism refers to the presence of multiple forms. Polymorphism usually occurs when there is a hierarchy of classes that are linked by inheritance.

C++ polymorphism means that depending on the type of object that invokes the function, a different function will be executed.

9. What do you mean by Call by Value and Call by Reference?

Ans. In this programming language to call a function we have 2 methods: Call by Value and Call by Reference

Call by Value	Call by Reference
A copy of a variable is passed.	A variable itself is passed fundamentally.
Calling a function by sending the values by copying variables.	Calling a function by sending the address of the passed variable.
The changes made in the function are never reflected outside the function on the variable. In short, the original value is never altered in Call by Value.	The changes made in the functions can be seen outside the function on the passed function. In short, the original value is altered in Call by reference.

10. What is the difference between function overloading and operator overloading?

Function Overloading

It is basically defining a function in numerous ways such that there are many ways to call it or in simple terms you have multiple versions of the same function

Parameterized Functions are a good example of Function Overloading as just by changing the argument or parameter of a function you make it useful for different purposes

Operator Overloading

It is basically giving practice of giving a special meaning to the existing meaning of an operator or in simple terms redefining the pre-redefined meaning

Polymorphism is a good example of an operator overloading as an object of allocations class can be used and called by different classes for different purposes

11. What is Function Overriding?

Ans. When a function of the same name, same arguments or parameters, and same return type already present/declared in the base class is used in a derived class is known as Function Overriding. It is an example of Runtime Polymorphism or Late Binding which means the overridden function will be executed at the run time of the execution.

12. What is virtual inheritance?

Ans. Virtual inheritance is a technique that ensures only one copy of a base class's member variables is inherited by grandchild-derived classes. Or in simple terms, virtual inheritance is used when we are dealing with a situation of multiple inheritances but want to prevent multiple instances of the same class from appearing in the inheritance hierarchy.

13. Explain the constructor in C++.

Ans. A constructor is a special type of member function of a class, whose name is the same as that of the class by whom it is invoked and initializes value to the object of a class.

There are 3 types of constructors:

- A. Default constructor:** It is the most basic type of constructor which accepts no arguments or parameters. Even if it is not called the compiler calls it automatically when an object is created.
- B. Parameterized constructor:** It is a type of constructor which accepts arguments or parameters. It has to be called explicitly by passing values in the arguments as these arguments help initialize an object when it is created. It also has the same name as that of the class. Also, It is used to overload constructors.
- C. Copy Constructor:** A copy constructor is a member function that initializes an object using another object of the same class. Also, the Copy constructor takes a reference to an object of the same class as an argument.

14. What are destructors in C++?

Ans. Destructors are members of functions in a class that delete an object when an object of the class goes out of scope. Destructors have the same name as the class preceded by a tilde (~) sign. Also, destructors follow a down-to-top approach, unlike constructors which follow a top-to-down.

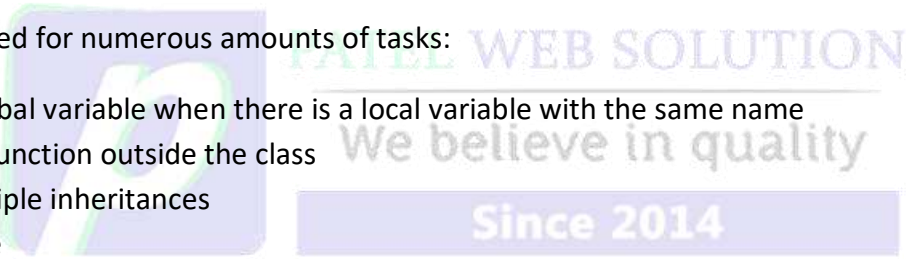
15. What is the purpose of the “delete” operator?

Ans. The delete operator is used to delete/remove all the characteristics/properties from an object by deallocating its memory; furthermore, it returns true or false in the end. In simple terms, it destroys or deallocates array and non-array(pointer) objects which are created by new expressions.

16. What does the Scope Resolution operator do?

Ans. A scope resolution operator is denoted by a '::' symbol. Just like its name this operator resolves the barrier of scope in a program. A scope resolution operator is used to reference a member function or a global variable out of their scope furthermore to which it can also access the concealed variable or function in a program.

Scope Resolution is used for numerous amounts of tasks:

- 
- A. To access a global variable when there is a local variable with the same name
 - B. To define the function outside the class
 - C. In case of multiple inheritances
 - D. For namespace

17. What are the C++ access modifiers?

Ans. The access restriction specified to the class members(whether it is member function or data member) is known as access modifiers/specifiers.

Access Modifiers are of 3 types:

- **Private** – It can neither be accessed nor be viewed from outside the class
- **Protected** – It can be accessed if and only if the accessor is the derived class
- **Public** – It can be accessed or be viewed from outside the class

18. What is a virtual function?

Ans. A member function in the base class redefined in a derived class is a virtual function. It is declared using the virtual keyword. It ensures that the correct function is called for an object, irrespective of the type of reference/pointer used for the function call. Virtual functions are mainly used for runtime polymorphism.

19. What is the C++ OOPs concept?

Ans. Concepts of OOPs:

- Object
- Class
- Inheritance
- Polymorphism
- Encapsulation
- Abstraction

Object: Anything that exists physically in the real world is called an object.

Class: The collection of objects is called class.

Inheritance: Properties of parent class inherited into child class is known as inheritance.

Polymorphism: It is the ability to exist in more than one form.

Encapsulation: Binding of code and data together into a single unit.

Abstraction: Hiding internal details and showing functionality to the user.

20. What is an inline function?

Ans. An inline function when called expands in line. When you call this function, the whole code of the inline function gets inserted or substituted at the inline function call.

Syntax:

Inline return-type function-name(parameters)

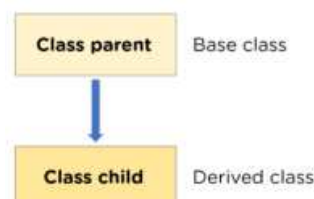
```
{
}
```

21. What is a copy constructor?

Ans. A copy function is a member function that uses another object from the same class to initialize a new thing. A copy function is, to put it simply, a function that produces an object by initializing it with a different object of the same class that has already been constructed.

22. What is inheritance?

Ans. Inheritance is the mechanism in which you can create a new class i.e. child class from the existing class i.e. parent class. This child class is also known as a derived class and the parent class is also called Base class.



23. What is the purpose of the “delete” operator?

Ans. The delete operator is used to delete/remove all the characteristics/properties from an object by deallocating its memory; furthermore, it returns true or false in the end. In simple terms, it destroys or deallocates array and non-array(pointer) objects which are created by new expressions.

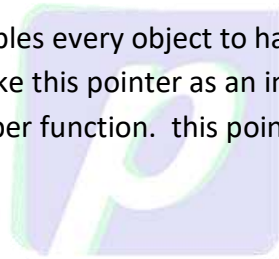
24. Define namespace in C++.

Ans. Namespaces enable us to organize named items that would otherwise have global scope into smaller scopes, allowing us to give them namespace scope. This permits program parts to be organized into distinct logical scopes with names. The namespace provides a place to define or declare identifiers such as variables, methods, and classes.

Or we could say that A namespace is a declarative zone that gives the identifiers (names of types, functions, variables, and so on) within it a scope. Namespaces are used to arrange code into logical categories and to avoid name clashes, which might happen when you have many libraries in your code base.

25. What is ‘this’ pointer in C++?

Ans. this pointer enables every object to have access to its own address through an essential pointer. All member functions take this pointer as an implicit argument. this pointer may be used to refer to the calling object within a member function. this pointer is used to pass an object as a parameter to another me.



www.vedantu.com
We believe in quality

Since 2014