

You are here: [Home](#) > [Interview Questions](#) > [C and C++ Interview Questions and Answers](#) > [C++ Interview Questions and Answers for Freshers, Experienced](#)

C++ Interview Questions and Answers for Freshers, Experienced

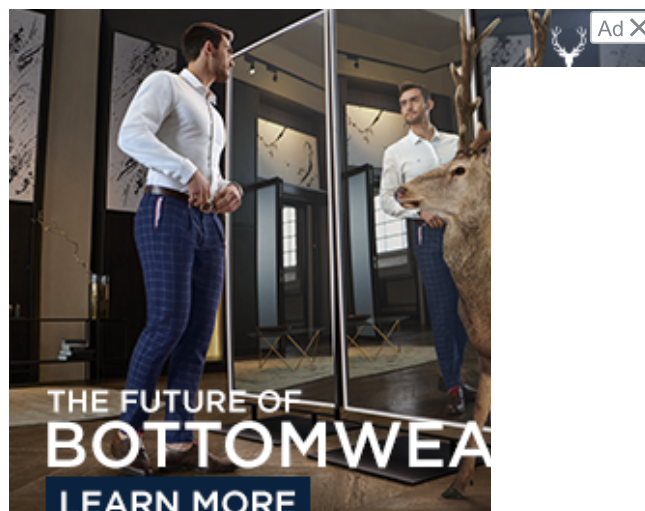
174 Votes

We are providing comprehensive set of C++ Interview Questions and Answers for Freshers and Experienced who wants to get into software industry. It's regardless of which domain or language you would like to pursue your career in, because 90% of campus placement interviews include questions from C or C++. The main reason being a candidate who is fresh out of college is not expected to be a master of widely used languages like Java, Python, Perl etc. and most of the universities have C or C++ as a part of their curriculum. For example, in most institutions C language is being taught along with data structure course and C++ as part of Object oriented programming. So always expect questions on C/C++ in the technical interviews for fresher.

One advantage with fresher interviews is that there are only limited number of concepts and questions and the same has been repeatedly being asked over years. If you want to pursue your career in system level programming like driver development, needless to say, you should show your mastery in C/C++ interviews. Here we have identified and collected a set of frequently asked C++ interview questions and answers for experienced and freshers as well. It's a continuously updated list, go through them, understand the concepts and excel in the interviews.

Ad X

1. What is difference between C and C++ ?



[This is a usual C or C++ interview question, mostly the first one you will face if you are fresher or appearing for campus interview. When answering this question please make sure you don't give the text book type explanations, instead give examples from real software scenario. Answer for this interview question can include below points, though its not complete list. This question itself can be a 1day interview !!!]

1. C++ is **Multi-Paradigm** (not pure OOP, supports both procedural and object oriented) while C follows procedural style programming.
2. In C data security is less, but in C++ you can use modifiers for your class members to make it inaccessible from outside.
3. C follows top-down approach (solution is created in step by step manner, like each step is processed into details as we proceed) but C++ follows a bottom-up approach (where base elements are established first and are linked to make complex solutions).
4. C++ supports function overloading while C does not support it.
5. C++ allows use of functions in structures, but C does not permit that.
6. **C++ supports reference variables** (two variables can point to same memory location). C does not support this.
7. C does not have a built in exception handling framework, though we can emulate it with other mechanism. **C++ directly supports exception handling**, which makes life of developer easy.

2. What is a class?

[Probably this would be the first question for a Java/c++ technical interview for freshers and sometimes for experienced as well. Try to give examples when you answer this question.]

Class defines a datatype, it's type definition of category of thing(s). But a class actually does not define the data, it just specifies the structure of data. To use them you need to create objects out of the class. Class can be considered as a blueprint of a building, you can not stay inside blueprint of building, you need to construct building(s) out of that plan. You can create any number of buildings from the blueprint, similarly you can create any number of objects from a class.

1. **class Vehicle**
2. {
3. **public:**
4. **int** numberOfTyres;
5. **double** engineCapacity;

```
6.      void drive(){  
7.          // code to drive the car  
8.      }  
9.  };
```

3. What is an Object/Instance?

Object is the instance of a class, which is concrete. From the above example, we can create instance of class Vehicle as given below

```
1.  Vehicle vehicleObject;
```

We can have different objects of the class Vehicle, for example we can have **Vehicle** objects with 2 tyres, 4tyres etc. Similarly different engine capacities as well.

4. What do you mean by C++ access specifiers ?

[Questions regarding access specifiers are common not just in c++ interview but for other object oriented language interviews as well.]

Access specifiers are used to define how the members (functions and variables) can be accessed outside the class. There are three access specifiers defined which are **public**, **private**, and **protected**

private:

Members declared as private are accessible only with in the same class and they cannot be accessed outside the class they are declared.

public:

Members declared as public are accessible from any where.

protected:

Members declared as protected can not be accessed from outside the class except a child class. This access specifier has significance in the context of inheritance.



Show schools your be

Ad Check out these steps
create a strong business s

GRE General Test

See the list.

5. What are the basics concepts of OOP?

[A must OOP / c++ interview question for freshers (some times asked in interviews for 1-2 years experienced also), which everybody answers as well. But the point is, it's not about the answer, but how you apply these OOPs concepts in real life. You should be able to give real life examples for each of these concepts, so prepare yourself with few examples before appearing for the interview. It has seen that even the experienced people get confused when it comes to the difference between basic OOP concepts, especially abstraction and encapsulation.]

Classes and Objects

Refer Questions 2 and 3 for the concepts about classes and objects

Encapsulation

Encapsulation is the mechanism by which data and associated operations/methods are bound together and thus hide the data from outside world. It's also called data hiding. In c++, encapsulation achieved using the access specifiers (private, public and protected). Data members will be declared as private (thus protecting from direct access from outside) and public methods will be provided to access these data. Consider the below class

```
1. class Person
2. {
3.     private:
4.         int age;
5.     public:
6.         int getAge(){
7.             return age;
8.         }
9.         int setAge(int value){
10.            if(value > 0){
11.                age = value;
12.            }
13.        }
14. };
```

In the class **Person**, access to the data field age is protected by declaring it as *private* and providing **public** access methods. What would have happened if there was no access methods and the field age was **public**? Anybody who has a **Person** object can set an invalid value (negative or very large value) for the age field. So by encapsulation we can preventing direct access from outside, and thus have complete control, protection and integrity of the data

Data abstraction

Data abstraction refers to hiding the internal implementations and show only the necessary details to the outside world. In C++ data abstraction is implemented using interfaces and abstract classes.

```

1.  class Stack
2.  {
3.      public:
4.          virtual void push(int)=0;
5.          virtual int pop()=0;
6.  };
7.
8.  class MyStack : public Stack
9.  {
10.     private:
11.         int arrayToHoldData[]; //Holds the data from stack
12.
13.     public:
14.         void push(int) {
15.             // implement push operation using array
16.         }
17.         int pop(){
18.             // implement pop operation using array
19.         }
20. };

```

In the above example, the outside world only need to know about the **Stack** class and its **push, pop** operations. Internally stack can be implemented using arrays or linked lists or queues or anything that you can think of. This means, as long as the push and pop method performs the operations work as expected, you have the freedom to change the internal implementation with out affecting other applications that use your Stack class.

Inheritance

Inheritance allows one class to inherit properties of another class. In other words, inheritance allows one class to be defined in terms of another class.

```

1.  class SymmetricShape

```

```
2. {
3.     public:
4.         int getSize()
5.         {
6.             return size;
7.         }
8.         void setSize(int w)
9.         {
10.            size = w;
11.        }
12.     protected:
13.         int size;
14. };
15.
16. // Derived class
17. class Square: public SymmetricShape
18. {
19.     public:
20.         int getArea()
21.         {
22.             return (size * size);
23.         }
24. };
```

In the above example, class **Square** inherits the properties and methods of class **SymmetricShape**. Inheritance is the one of the very important concepts in C++/OOP. It helps to modularise the code, improve reusability and reduces tight coupling between components of the system.



Show schools your be

Ad Check out these steps
create a strong business s

GRE General Test

See the list.

6. What is the use of volatile keyword in c++? Give an example.

Most of the times compilers will do optimization to the code to speed up the program. For

example in the below code,

```
1. int a = 10;  
2. while( a == 10){  
3.     // Do something  
4. }
```

compiler may think that value of 'a' is not getting changed from the program and replace it with 'while(true)', which will result in an infinite loop. In actual scenario the value of 'a' may be getting updated from outside of the program.

Volatile keyword is used to tell compiler that the variable declared using volatile may be used from outside the current scope so that compiler wont apply any optimization. This matters only in case of multi-threaded applications.

In the above example if variable 'a' was declared using volatile, compiler will not optimize it. In shot, value of the volatile variables will be read from the memory location directly.

Page 1 of 4

Pages

1

2

3

4

New Articles



Best Job interview tips to land you in your dream job

2015-08-09

Popular Articles



SQL Queries Interview Questions and Answers

2013-02-26



Database & SQL Interview Questions

2013-03-02



C# Interview Questions

2013-03-17



Java Interview Questions

2013-08-13



HR Interview Questions for Freshers

2014-08-24

Popular Videos



How to improve your Interview, Salary Negotiation, Communication & Presentation Skills.

Got a tip or Question?
Let us know

Related Articles

[Advanced C Interview Questions and Answers for Experienced](#)

[Embedded C Interview Questions and Answers](#)

[C Interview Questions and Answers](#)

Site Links

[Careers Jobs](#)

[Interview Questions](#)

[Placement Test](#)

[Seminar Topics](#)

[B.E Projects](#)

Community

[Write for us](#)

[Blog](#)

[Forum](#)

[Contribute Projects & Seminars](#)

[Report site issue](#)

We're on Social

[Facebook](#)

[Twitter](#)

[Linkedin](#)

[YouTube](#)

[Google+](#)

Subscribe to our Newsletter

Subscribe to our weekly Newsletter and receive updates via email.

[Subscribe](#)

Copyright © 2016 A4Academics. All rights reserved.

[Privacy Policy & Site Terms](#)

[About us](#) | [Contact us](#) | [Advertise](#)