

C++ Template Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on C++ Template will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [C++ Template Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in C++ Template category. To ensure quality, each submission is checked by our team, before it becomes live. This [C++ Template Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



C++ Template Interview Questions And Answers Guide.

Question - 1:

Why we use :: template-template parameter?

- a) binding
- b) rebinding
- c) both a & b
- d) none of these

Ans:

- c) both a & b

[View All Answers](#)

Question - 2:

Which of the things does not require instantiation?

- a) functions
- b) non virtual member function
- c) member class
- d) all of the mentioned

Ans:

- d) all of the mentioned

[View All Answers](#)

Question - 3:

What is the output of this program?

```
#include <iostream>
using namespace std;
template <class T>
T max (T a, T b)
{
    return (a>b?a:b);
}
int main ()
{
    int i = 5, j = 6, k;
    long l = 10, m = 5, n;
    k = max(i, j);
    n = max(l, m);
    cout << k << endl;
    cout << n << endl;
    return 0;
}
```

- a) 6
- b) 6
- 10
- c) 5
- 10
- d) 6
- 5

Ans:

- b) 6
- 10

[View All Answers](#)

Question - 4:

Which parameter is legal for non-type template?



- a) pointer to member
- b) object
- c) class
- d) none of the mentioned

Ans:

- a) pointer to member

[View All Answers](#)

Question - 5:

What is the output of this program?

```
#include <iostream>
using namespace std;
template <typename T, int count>
void loopIt(T x)
{
    T val[count];
    for(int ii = 0; ii < count; ii++)
    {
        val[ii] = x++;
        cout << val[ii] << endl;
    }
};
int main()
{
    float xx = 2.1;
    loopIt<float, 3>(xx);
}
```

- a) 2.1
- b) 3.1
- c) 4.1
- d) 2.1
3.1
4.1

Ans:

- d) 2.1
3.1
4.1

[View All Answers](#)

Question - 6:

What is the output of this program?

```
#include <iostream>
using namespace std;
template <class type>
class Test
{
public:
    Test()
    {
    };
    ~Test()
    {
    };
    type Funct1(type Var1)
    {
        return Var1;
    }
    type Funct2(type Var2)
    {
        return Var2;
    }
};
int main()
{
    Test<int> Var1;
    Test<double> Var2;
    cout << Var1.Funct1(200);
    cout << Var2.Funct2(3.123);
    return 0;
}
```

- a) 100
- b) 200
- c) 3.123
- d) 200 3.123

Ans:

- d) 200 3.123



[View All Answers](#)

Question - 7:

What is the output of this program?

```
#include <iostream>
using namespace std;
template <class T, int N>
class mysequence
{
    T memblock [N];
public:
    void setmember (int x, T value);
    T getmember (int x);
};
template <class T, int N>
void mysequence<T,N> :: setmember (int x, T value)
{
    memblock[x] = value;
}
template <class T, int N>
T mysequence<T,N> :: getmember (int x)
{
    return memblock[x];
}
int main ()
{
    mysequence <int, 5> myints;
    mysequence <double, 5> myfloats;
    myints.setmember (0, 100);
    myfloats.setmember (3, 3.1416);
    cout << myints.getmember(0) << 'n';
    cout << myfloats.getmember(3) << 'n';
    return 0;
}
```

- a) 100
- b) 3.1416
- c) 100
3.1416
- d) none of the mentioned

Ans:

- c) 100

[View All Answers](#)

Question - 8:

What is the validity of template parameters?

- a) inside that block only
- b) inside the class
- c) whole program
- d) any of the mentioned

Ans:

- a) inside that block only

[View All Answers](#)

Question - 9:

Which keyword can be used in template?

- a) class
- b) typename
- c) both a & b
- d) function

Ans:

- c) both a & b

[View All Answers](#)

Question - 10:

What is meant by template parameter?

- a) It can be used to pass a type as argument
- b) It can be used to evaluate a type.
- c) It can of no return type
- d) None of the mentioned

Ans:

- a) It can be used to pass a type as argument

[View All Answers](#)



Question - 11:

Can you please explain the difference between a template class and class template?

Ans:

Template class: A class that has generic definition or a class with parameters which is not instantiated until the information is provided by the client. It is referred to a jargon for plain templates.

Class template: The individual construction of a class is specified by a class template which is almost similar the way how individual objects are constructed by using a class. It is referred to a jargon for plain classes.

[View All Answers](#)

Question - 12:

What is Class template?

Ans:

The individual construction of a class is specified by a class template which is almost similar the way how individual objects are constructed by using a class. It is referred to a jargon for plain classes.

[View All Answers](#)

Question - 13:

What is Template class?

Ans:

A class that has generic definition or a class with parameters which is not instantiated until the information is provided by the client. It is referred to a jargon for plain templates.

[View All Answers](#)

Question - 14:

Please tell me how is static data member similar to a global variable?

Ans:

The life of a static data member exists between the functions which means that they are resident through out the execution of a program like the global variables.

[View All Answers](#)

Question - 15:

Tell us what is the STL, standard template library?

Ans:

The Standard Template Library, or STL, is a C++ library of container classes, algorithms, and iterators; it provides many of the basic algorithms and data structures. The STL includes the classes vector, list, deque, set, multiset, map, multimap, hash_set, hash_multiset, hash_map, and hash_multimap.

[View All Answers](#)

Question - 16:

Do you know what is class using C++?

Ans:

A class holds the data and functions that operate on the data. It serves as the template of an object.

[View All Answers](#)

Question - 17:

Tell me what are the syntax and semantics for a function template?

Ans:

Templates is one of the features of C++. Using templates, C++ provides a support for generic programming.

We can define a template for a function that can help us create multiple versions for different data types.

A function template is similar to a class template and its syntax is as follows:

```
template <class T>
Return-type functionName (arguments of type T)
{
    //Body of function with type T wherever appropriate
}
```

[View All Answers](#)

Question - 18:

Can you please explain what are the characteristics of Object Oriented programming language?

Ans:

Some key features of the Object Oriented programming are:

- Emphasis on data rather than procedure
- Programs are divided into entities known as objects
- Data Structures are designed such that they characterize objects
- Functions that operate on data of an object are tied together in data structures
- Data is hidden and cannot be accessed by external functions
- Objects communicate with each other through functions



New data and functions can be easily added whenever necessary
Follows bottom up design in program design

[View All Answers](#)

Question - 19:

Tell me what are the basic Concepts used in the Object-Oriented Programming language?

Ans:

Object
Class
Data Abstraction and Encapsulation
Polymorphism
Inheritance
Message passing
Dynamic binding

[View All Answers](#)

Question - 20:

What is Polymorphism in C++?

Ans:

Polymorphism enables one common interface for many implementations, and for objects to act differently under different circumstances. You can also achieve polymorphism in C++ by function overloading, operator overloading and implementation inheritance.

[View All Answers](#)

Question - 21:

What is Encapsulation in C++?

Ans:

The wrapping up of data and member function into an object is called encapsulation. The data is not accessible to the outside world and only those functions which are wrapped into the object can access it. An encapsulated objects act as a "black box" for other parts of the program which interact with it. They provide a service, but the calling objects do not need to know the details how the service is accomplished.

[View All Answers](#)

Question - 22:

What is Inheritance in C++?

Ans:

Inheritance enables a new class to reuse the state and behavior of old class. The new class inherits properties and methods from the old class and is called as derived class and the old class is called as base class. The methods thus inherited can be extended using overriding facility of C++.

[View All Answers](#)

Question - 23:

What is Class element in C++?

Ans:

A class is a user defined data type. It serves as a template of the objects. You can define structure and behavior of an object using class. It includes data and the member functions that operate on data.

[View All Answers](#)

C++ Most Popular & Related Interview Guides

- 1 : [C++ Pointers & Functions Interview Questions and Answers.](#)
- 2 : [C++ Operator Overloading Interview Questions and Answers.](#)
- 3 : [C++ Exception Handling Interview Questions and Answers.](#)
- 4 : [C++ Friend Interview Questions and Answers.](#)
- 5 : [C++ Virtual Functions Interview Questions and Answers.](#)
- 6 : [C++ Constructors Interview Questions and Answers.](#)
- 7 : [C++ Type Checking Interview Questions and Answers.](#)
- 8 : [C++ Inheritance Interview Questions and Answers.](#)
- 9 : [C++ Access Control Interview Questions and Answers.](#)
- 10 : [C++ Inline Function Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)