C++ Friend 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++ Friend 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++ Friend 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++ Friend category. To ensure quality, each submission is checked by our team, before it becomes live. This C++ Friend 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++ Friend Interview Questions And Answers Guide.

Question - 1:

List the advantages of using friend classes?

Ans:

There are situations when private data members need to be accessed and used by 2 classes simultaneously. In these kind of situations we can introduce friend functions which have an access to the private data members of both the classes. Friend functions need to be declared as 'friend' in both the classes. They need not be members of either of these classes.

View All Answers

Question - 2:

What are the C++ friend classes? Explain its uses with an example?

Ans:

A friend class and all its member functions have access to all the private members defined within other class.

```
#include <iostream>
using namespace std;
class Numbers
{
   int a;
   int b;
   public:
        Numbers(int i, int j)
        {
              a = i;
              b = j;
        }
        friend class Average;
};
class Average
{
     public:
          int average(Numbers x);
};
int Average:: average(Numbers x)
{
        return ((x.a + x.b)/2);
}
int main()
{
       Numbers ob(23, 67);
       Average avg;
       cout<<"The average of the numbers is: "<<avg.average(ob);
       return 0;
}
```

Note the friend class member function average is passed the object of Numbers class as parameter.

View All Answers

Question - 3:

List the characteristics of friend functions?

Ans:

Friend functions are not a part of the class and are external. This function is a "Friend" of a class. This is to say, it has special privileges to access to the class's private and protected members.

View All Answers

Question - 4:

What is the friend class in C++?

Ans:

When a class declares another class as its friend, it is giving complete access to all its data and methods including private and protected data and methods to the friend class member methods. Friendship is one way only, which means if A declares B as its friend it does NOT mean that A can access private data of B. It only means that B can access all data of A.

View All Answers

Question - 5:

Pick out the correct statement.
a) A friend function may be a member of another class.
b) A friend function may not be a member of another class.
c) A friend function may or may not be a member of another class.
d) None of the mentioned

Ans:

c) A friend function may or may not be a member of another class.

View All Answers

Question - 6:

Where does keyword 'friend' should be placed?
a) function declaration
b) function definition
c) main function
d) None of the mentioned

Ans:

a) function declaration
Explanation:
The keyword friend is placed only in the function declaration of the friend function and not in the function definition because it is used toaccess the member of a class.

View All Answers

Question - 7:

What is the output of this program?

```
#include <iostream>
using namespace std;
class base
{
   int val1, val2;
   public:
   int get()
{
      val1 = 100;
      val2 = 300;
}
   friend float mean(base ob);
};
float mean(base ob)
{
   return float(ob.val1 + ob.val2) / 2;
}
int main()
{
   base obj;
   obj.get();
   cout << mean(obj);
   return 0;
}
```

a) 200
b) 150
c) 100
d) 300

Ans:

a) 200

View All Answers

Question - 8:

What is output of this program?

```
#include <iostream>
using namespace std;
class sample
{
   private:
   int a, b;
   public:
```

```
    void test()
    {
      a = 100;
      b = 200;
    }
    friend int compute(sample e1);
};
int compute(sample e1)
{
    return int(e1.a + e1.b) - 5;
}
int main()
{
    sample e;
    e.test();
    cout  << compute(e);
    return 0;
}
```

a) 100
b) 200
c) 300
d) 295

Ans:

d) 295
Explanation:
In this program, we are finding a value from the given function by using the friend for compute function.
Output:

```
$ g++ friend4.cpp
$ a.out
295
```

View All Answers

Question - 9:

What is the output of this program?

```
#include <iostream>
using namespace std;
class sample;
class sample1
{
    int width, height;
    public:
    int area ()
    {
      return (width * height);}
      void convert (sample a);
    };
class sample
{
    private:
    int side;
    public:
    void set_side (int a)
    {
      side = a;
    }
    friend class sample1;
};
void sample1::convert (sample a)
{
    width = a.side;
    height = a.side;
}
int main ()
{
    sample sqr;
    sample1 rect;
    sqr.set_side(6);
    rect.convert(sqr);
    cout << rect.area();
    return 0;
}
```

a) 24
b) 35
c) 16
d) 36

Ans:

d) 36
Explanation:
In this program, we are using the friend for the class and calculating the area of the square.

Output:
$ g++ friend2.cpp
$ a.out
36

View All Answers

Question - 10:

What is the output of this program?

```
#include <iostream>
using namespace std;
class sample
{
   int width, height;
   public:
   void set_values (int, int);
   int area () {return (width * height);}
   friend sample duplicate (sample);
};
void sample::set_values (int a, int b)
{
   width = a;
   height = b;
}
sample duplicate (sample rectparam)
{
   sample rectres;
   rectres.width = rectparam.width * 2;
   rectres.height = rectparam.height * 2;
   return (rectres);
}
int main ()
{
   sample rect, rectb;
   rect.set_values (2, 3);
   rectb = duplicate (rect);
   cout << rectb.area();
   return 0;
}
```

a) 20
b) 16
c) 24

Ans:

c) 24
Explanation:
In this program, we are using the friend function for duplicate function and calculating the area of the rectangle.
Output:
$ g++ friend1.cpp
$ a.out
24

View All Answers

Question - 11:

What is the output of this program?

```
#include <iostream>
using namespace std;
class Box
{
   double width;
   public:
   friend void printWidth( Box box );
   void setWidth( double wid );
};
void Box::setWidth( double wid )
{
   width = wid;
}
void printWidth( Box box )
{
   box.width = box.width * 2;
   cout << "Width of box : " << box.width << endl;
}
int main( )
{
   Box box;
   box.setWidth(10.0);
   printWidth( box );
   return 0;
}
```

a) 40

b) 5
c) 10
d) 20

Ans:

d) 20
Explanation:
We are using the friend function for printwidth and multiplied the width value by 2, So we got the output as 20
Output:
$ g++ friend.cpp
$ a.out
20

View All Answers

Question - 12:

Which keyword is used to declare the friend function?
a) firend
b) friend
c) classfriend
d) myfriend

Ans:

b) friend

View All Answers

Question - 13:

What is the syntax of friend function?
a) friend class1 Class2;
b) friend class;
c) friend class
d) None of the mentioned

Ans:

a) friend class1 Class2;
Explanation:
In option a, the class2 is the friend of class1 and it can access all the private and protected members of class1.

View All Answers

Question - 14:

Which rule will not affect the friend function?
a) private and protected members of a class cannot be accessed from outside
b) private and protected member can be accessed anywhere
c) both a & b
d) None of the mentioned

Ans:

a) private and protected members of a class cannot be accessed from outside
Explanation:
Friend is used to access private and protected members of a class from outside the same class.

View All Answers

Question - 15:

What are the advantages of using friend classes?

Ans:

There are situations when private data members need to be accessed and used by 2 classes simultaneously. In these kind of situations we can introduce friend functions which have an access to the private data members of both the classes. Friend functions need to be declared as â€˜friend' in both the classes. They need not be members of either of these classes.

View All Answers

Question - 16:

What is a friend function?

Ans:

To allow a non-member function the access to private members of a class, it needs to be friend of that class. Friend functions can access private and protected data of the class. To make a non-member function friend of a class, its declaration needs to be made inside the class and it has to be preceded by the keyword friend. We can also have a member function of a class to be friend of certain other class. Even if a function is not a member of any class, it can still be friend of multiple classes.
If we write equivalent friend function for a member function, then friend function has one extra parameter because being a non-member of the class, it does not have the caller object. Therefore, it does not have this pointer.
Friend functions are very useful for overloading certain types of operators. They also make creation of some type of I/O functions easier.
Consider following example of class 3D having data members x, y and z. Overloaded binary operator * for scalar multiplication. It should work in both cases:
ob1 = ob2 * 3;
ob1 = 3 * ob2;
Note that first can be achieved through member or friend function. But for the second case we need to write friend function since there is no caller object.
class 3D
{

```
    int x, y, z;
    public:
        3D (int a=0, int b=0, int c=0)
        {
            x = a;
            y = b;
            z = c;
        }
        3D show()
        {
            cout<<â€•The elements are:
â€•
            cout<<â€•x:â€•<<this->x<<â€•, y:<<this->y <<â€•, z:â€•<<this->z;
        }
         friend 3D operator * (3D, int);
         friend 3D operator * (int, 3D);
};
3D operator * (3D ob, int i) //friend function's definition is written outside class
{
      3D tob;
      tob.x = ob.x * i;
      tob.y = ob.y * i;
      tob.z = ob.z * i;
      return tob;
}
3D operator * (int i, 3D ob) //friend function's definition is written outside class
{
      3D tob;
      tob.x = ob.x * i;
      tob.y = ob.y * i;
      tob.z = ob.z * i;
      return tob;
}
int main()
{
  3D pt1(2,-4,5), pt2, pt3;
  pt 2 = pt1 * 3;
  pt3 = -2 * pt1
  cout << â€œ
 Point one's dimensions are:
â€œ<<pt1.show();
  cout << â€œ
 Point two's dimensions are:
â€œ<<pt2.show();
  cout << â€œ
 Point three's dimensions are:
â€œ<<pt3.show();
  return 0;
}
```

The o/p would be:
Point one's dimensions are:
x:2, y:-4, z:5
Point two's dimensions are:
x:6, y:-12, z:15
Point three's dimensions are:
x:-4, y:8, z:-10
Thus, friend functions are useful when we have to overload operators which have no caller object.

View All Answers

Question - 17:

Explain the characteristics of friend functions?

Ans:

A friend function is not in the scope of the class n which it has been declared as friend.
It cannot be called using the object of that class.
It can be invoked like a normal function without any object.
Unlike member functions, it cannot use the member names directly.
It can be declared in public or private part without affecting its meaning.
Usually, it has objects as arguments.

View All Answers

Question - 18:

Explain advantages of using friend classes?

Ans:

A friend function has the following advantages:
- Provides additional functionality which is kept outside the class.
- Provides functions that need data which is not normally used by the class.
- Allows sharing private class information by a non member function.

View All Answers

Question - 19:

What are friend classes?

Ans:

The friend function is a â€˜non member function' of a class. It can access non public members of the class. A friend function is external to the class definition.

View All Answers

Question - 20:

Explain friend function?

Ans:

When the application is needed to access a private member of another class, the only way is to utilize the friend functions. The following are the guidelines to handle friend functions.

- Declare the function with the keyword â€˜friend' that is followed by return type and followed by the function name.
- Specify the class whose private members are to be accessed in the friend function within parenthesis of the friend function.
- Write the code of the friend function in the class.

The following code snippet illustrates the use of friend function:

```
friend void display(car); //Friend of the class 'car'
void display(car myCar)
{
    cout<<"
The color of my car is : "<<myCar.color;
    cout<<"
The speed of my car is : "<<myCar.speed;
}
```

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++ Template 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