

# **C++ Syntax 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++ Syntax 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++ Syntax 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++ Syntax category. To ensure quality, each submission is checked by our team, before it becomes live. This [C++ Syntax 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](http://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](https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG)



## C++ Syntax Interview Questions And Answers Guide.

### Question - 1:

What is C strings syntax?

#### Ans:

Strings are arrays of chars. String literals are words surrounded by double quotation marks.

"This is a static string"

To declare a string of 49 letters, you would want to say:

```
char string[50];
```

[View All Answers](#)

### Question - 2:

How to access a variable of the structure?

#### Ans:

To access a variable of the structure it goes:

```
name_of_single_structure.name_of_variable;
```

For example:

```
struct example {  
    int x;  
};
```

```
struct example an_example; //Treating it like a normal variable type
```

```
an_example.x = 33; //How to access its members
```

[View All Answers](#)

### Question - 3:

What is format for defining a structure?

#### Ans:

The format for defining a structure is:

```
struct Tag {  
    Members  
};
```

Where Tag is the name of the entire type of structure and Members are the variables within the struct. To actually create a single structure the syntax is

```
struct Tag name_of_single_structure;
```

[View All Answers](#)

### Question - 4:

What is general format for a prototype?

#### Ans:

The general format for a prototype is simple:

```
return-type function_name ( arg_type arg1, ..., arg_type argN );
```

arg\_type just means the type for each argument -- for instance, an int, a float, or a char. It's exactly the same thing as what you would put if you were declaring a variable.

[View All Answers](#)

### Question - 5:

What is prototype for that C string function?

#### Ans:

The prototype for that function is:

```
istream& getline(char *buffer, int length, char terminal_char);
```

The char \*buffer is a pointer to the first element of the character array, so that it can actually be used to access the array. The int length is simply how long the string to be input can be at its maximum (how big the array is). The char terminal\_char means that the string will terminate if the user inputs whatever that character is.



Keep in mind that it will discard whatever the terminal character is.

It is possible to make a function call of `cin.getline(array, 50)`; without the terminal character. Note that 'n' is the way of actually telling the compiler you mean a new line, i.e. someone hitting the enter key.

For an example:

```
#include <iostream>
using namespace std;
int main()
{
    char string[256]; // A nice long string
    cout<<"Please enter a long string: ";
    cin.getline ( string, 256, 'n' ); // Input goes into string
    cout<<"Your long string was: "<< string <<endl;
    cin.get();
}</endl;
</iostream>
```

[View All Answers](#)

### Question - 6:

How one would use switch in a program?

**Ans:**

```
#include <iostream>
using namespace std;
void playgame()
{
    cout << "Play game called";
}
void loadgame()
{
    cout << "Load game called";
}
void playmultiplayer()
{
    cout << "Play multiplayer game called";
}

int main()
{
    int input;

    cout<<"1. Play gamen";
    cout<<"2. Load gamen";
    cout<<"3. Play multiplayer";
    cout<<"4. Exitn";
    cout<<"Selection: ";
    cin>> input;
    switch ( input ) {
    case 1: // Note the colon, not a semicolon
        playgame();
        break;
    case 2: // Note the colon, not a semicolon
        loadgame();
        break;
    case 3: // Note the colon, not a semicolon
        playmultiplayer();
        break;
    case 4: // Note the colon, not a semicolon
        cout<<"Thank you for playing!\n";
        break;
    default: // Note the colon, not a semicolon
        cout<<"Error, bad input, quitting";
        break;
    }
    cin.get();
}</iostream>
```

[View All Answers](#)

### Question - 7:

What is switch case in C++ Syntax?

**Ans:**

Switch case statements are a substitute for long if statements that compare a variable to several "integral" values ("integral" values are simply values that can be expressed as an integer, such as the value of a char). The basic format for using switch case is outlined below. The value of the variable given into switch is compared to the value following each of the cases, and when one value matches the value of the variable, the computer continues executing the program from that point.

```
switch ( <variable> ) {
case this-value:
    Code to execute if <variable> == this-value
    break;
case that-value:
    Code to execute if <variable> == that-value
```



```
break;
...
default:
    Code to execute if <variable> does not equal the value following any of the cases
    break;
}</variable></variable></variable></variable>
```

[View All Answers](#)

### Question - 8:

How to defines the function in C++?

#### Ans:

When the programmer actually defines the function, it will begin with the prototype, minus the semi-colon. Then there should always be a block with the code that the function is to execute, just as you would write it for the main function. Any of the arguments passed to the function can be used as if they were declared in the block. Finally, end it all with a cherry and a closing brace. Okay, maybe not a cherry.

Let's look at an example program:

```
#include <iostream>
using namespace std;
int mult ( int x, int y );
int main()
{
    int x;
    int y;

    cout<<"Please input two numbers to be multiplied: ";
    cin>> x >> y;
    cin.ignore();
    cout<<"The product of your two numbers is "<< mult ( x, y ) <<"n";
    cin.get();
}
int mult ( int x, int y )
{
    return x * y;
}</iostream>
```

[View All Answers](#)

### Question - 9:

What is functions Syntax in C++?

#### Ans:

Functions that a programmer writes will generally require a prototype. Just like a blueprint, the prototype tells the compiler what the function will return, what the function will be called, as well as what arguments the function can be passed. When I say that the function returns a value, I mean that the function can be used in the same manner as a variable would be. For example, a variable can be set equal to a function that returns a value between zero and four.

For example:

```
#include <cstdlib> // Include rand()
using namespace std; // Make rand() visible
int a = rand(); // rand is a standard function that all compilers have</cstdlib>
```

[View All Answers](#)

### Question - 10:

What is do..while loops structure?

#### Ans:

DO..WHILE loops are useful for things that want to loop at least once. The structure is

```
do {
} while ( condition );
```

[View All Answers](#)

### Question - 11:

What is while loops?

#### Ans:

while ( condition ) { Code to execute while the condition is true } The true represents a boolean expression which could be  $x == 1$  or  $x != 7$  (x does not equal 7). It can be any combination of boolean statements that are legal. Even,  $(x == 5 \parallel v == 7)$  which says execute the code while x equals five or while v equals 7. Notice that a while loop is the same as a for loop without the initialization and update sections. However, an empty condition is not legal for a while loop as it is with a for loop.

Example:

```
#include <iostream>
using namespace std; // So we can see cout and endl
int main()
{
    int x = 0; // Don't forget to declare variables

    while ( x < 10 ) { // While x is less than 10
        cout<< x <<endl;
        x++; // Update x so the condition can be met eventually
    }
```



```
cin.get();
}</endl;
</iostream>
```

[View All Answers](#)

### Question - 12:

What is The syntax for a for loop?

**Ans:**

```
for ( variable initialization; condition; variable update ) {
    Code to execute while the condition is true
}
```

[View All Answers](#)

### Question - 13:

What is conditions when using boolean operators?

**Ans:**

A. !( 1 || 0 )      ANSWER: 0  
B. !( 1 || 1 && 0 )      ANSWER: 0 (AND is evaluated before OR)  
C. !( ( 1 || 0 ) && 0 )      ANSWER: 1 (Parenthesis are useful)

[View All Answers](#)

### Question - 14:

What is Else If Syntax?

**Ans:**

```
if ( <condition> ) {
    // Execute these statements if <condition> is TRUE
}
else if ( <another condition=""> ) {
    // Execute these statements if <another condition=""> is TRUE and
    // <condition> is FALSE
}</condition></another></another></condition></condition>
```

[View All Answers](#)

### Question - 15:

What is Else Syntax in C++?

**Ans:**

It can look like this:

```
if ( TRUE ) {
    // Execute these statements if TRUE
}
else {
    // Execute these statements if FALSE
}
```

[View All Answers](#)

### Question - 16:

What is basic if statement syntax?

**Ans:**

The structure of an if statement is as follows:

```
if ( TRUE )
```

Execute the next statement

Here is a simple example that shows the syntax:

```
if ( 5 < 10 )
cout<<"Five is now less than ten, that's a big surprise";
```

[View All Answers](#)

### Question - 17:

What relational operators if statements in C++?

**Ans:**

Here are the relational operators, as they are known, along with examples:

```
> greater than      5 > 4 is TRUE
< less than      4 < 5 is TRUE
>= greater than or equal      4 >= 4 is TRUE
<= less than or equal      3 <= 4 is TRUE
== equal to      5 == 5 is TRUE
!= not equal to      5 != 4 is TRUE
```

[View All Answers](#)



### Question - 18:

How to demonstrate the use of a variable?

#### Ans:

Here is a sample program demonstrating the use of a variable:

```
#include <iostream>

using namespace std;
int main()
{
    int thisisanumber;
    cout<<"Please enter a number: ";
    cin>> thisisanumber;
    cin.ignore();
    cout<<"You entered: "<< thisisanumber <<"\n";
    cin.get();
}</iostream>
```

[View All Answers](#)

### Question - 19:

How to Declaring Variables in C++?

#### Ans:

To declare a variable you use the syntax "type <name>;". Here are some variable declaration examples:

```
1- int x;
2- char letter;
3- float the_float;
```

It is permissible to declare multiple variables of the same type on the same line; each one should be separated by a comma.

```
1- int a, b, c, d;
```

If you were watching closely, you might have seen that declaration of a variable is always followed by a semicolon (note that this is the same procedure used when you call a function).</name>

[View All Answers](#)

### Question - 20:

Write a short code using C++ to print out all odd number from 1 to 100 using a for loop?

#### Ans:

```
for( unsigned int i = 1; i <= 100; i++ )
if( i & 0x00000001 )
    cout << i << ", ";
```

[View All Answers](#)

### Question - 21:

What OO language is best?

#### Ans:

Whichever one works best for the organization.

We believe in honesty, not advocacy, and the honest answer is that there is no single answer. What is the organization's policy regarding languages? Must there be one and only one official language? What is the skill level of the staff? Are they gung-ho developers with advanced degrees in computer science/engineering or people who understand the business and have survival skills in software? Do they already have OO skills? In which language(s)? What sort of software development is being done: extending someone else's framework or building from scratch for resale? What sort of performance constraints does the software have? Is it space constrained or speed constrained? If speed, is it typically bound by I/O, network, or CPU? Regarding libraries and tools, are there licensing considerations? Are there strategic partnership relationships that affect the choice of languages? Many of these questions are nontechnical, but they are the kind of questions that need to be answered the "which language" issue can be addressed.

Regarding the choice between C++ and Java, java is a simpler language and thus it is generally easier to use. However C++ is more established and allows finer control over resources (for example, memory management), and this is required for some applications. Also, C++ has language features such as destructors

[View All Answers](#)

### Question - 22:

What's the "Software Peter Principle"?

#### Ans:

The Software Peter Principle is in operation when unwise developers "improve" and "generalize" the software until they themselves can no longer understand it, then the project slowly dies.

The Software Peter Principle can ruin projects. The insidious thing about the Software Peter Principle is that it's a silent killer - by the time the symptoms are visible, the problem has spread throughout every line of code in the project.

Foolish managers deal with symptoms rather than prevention, and they think everything is okay unless there are visible bugs. Yet the problem isn't bugs, at least initially. The problem is that the project is collapsing under its own weight.



The best way to avoid this problem is to build to the skill level of the maintainers, not of the developers. If the typical maintainer won't understand the software then it's simply too complex for the organization to maintain. This means avoiding tricky, sophisticated, subtle, clever techniques unless there is a compelling reason for them. Cleverness is evil; use it only when necessary.

Shown concern for the long-term health of the system being developed.

[View All Answers](#)

### Question - 23:

What is the most common mistake on C++ and OO projects?

#### Ans:

Unnecessary complexity - the plague of OO technology.

Complexity, like risk, is a fact of life that can't be avoided. Some software systems have to be complex because the business processes they represent are complex. But unfortunately many intermediate developers try to "make things better" by adding generalization and flexibility that no one has asked for or will ever need. The customer wants a cup of tea, and the developers build a system that can boil the ocean [thanks to John Vlissides for this quip]. The result is unnecessary complexity, which increases the risk of failure. The intentions might be good but the result can be deadly.

Here are a few guidelines.

• Don't solve problems that don't need to be solved.

• Don't worry about the future until you're sure you can survive the present.

39

• Don't build things for the fun of it.

• The organization's health is more important than the developer's desire to play with the latest whiz-bang tool or technique.

• Don't add risk without a compelling and measurable benefit to the project.

• Don't invest in the future if your current project is in trouble.

Avoid the "death by one thousands cut" syndrome by avoiding unnecessary complexity.

[View All Answers](#)

### Question - 24:

Explain what are the Sizes and ranges of the Basic C++ data types?

#### Ans:

Following table is with respect to a 16-bit word machine:

Type	Bytes	Range
char	1	-128 to 127
unsigned char	1	0 to 255
signed char	1	-128 to 127
int	2	-32768 to 32767
unsigned int	2	0 to 65535
signed int	2	-32768 to 32767
short int	2	-32768 to 32767
unsigned short int	2	0 to 65535
signed short int	2	-32768 to 32767
long int	4	-2147483648 to 2147483647
signed long int	4	-2147483648 to 2147483647
unsigned long int	4	0 to 4294967295
float	4	3.4E-38 to 3.4E+38
double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

[View All Answers](#)

### Question - 25:

Do you know what are the new features that ISO/ANSI C++ has added to original C++ specifications?

#### Ans:

New Data types:

- bool

- wchar\_t

New operators

- const\_cast

- static\_cast

- dynamic\_cast

- reinterpret\_cast

- typeid

Class implementation

- Explicit constructors

- Mutable members

Namespace scope

Operator keywords

New keywords

New headers

[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++ Friend Interview Questions and Answers.](#)
- 6 : [C++ Virtual Functions Interview Questions and Answers.](#)
- 7 : [C++ Constructors Interview Questions and Answers.](#)
- 8 : [C++ Type Checking Interview Questions and Answers.](#)
- 9 : [C++ Inheritance Interview Questions and Answers.](#)
- 10 : [C++ Access Control Interview Questions and Answers.](#)

**Follow us on FaceBook**

[www.facebook.com/InterviewQuestionsAnswers.Org](http://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)