Share on your Social Media

Top 20 C and C++ Interview Questions and Answers

Published On: May 23, 2022

C and C++ Interview Questions and Answers

C and C++ programming languages are in demand in industries for operating system development and application development. Here are the popular and frequently asked **top 20 C and C++ interview questions and answers** to help you ace the technical rounds.

[Download C and C++ Interview Questions PDF]

C and C++ Interview Questions and Answers for Freshers

1. What is C programming?

Related Courses at SLA

- C C++ Training in OMR
- C C++ Training in Chennai

Related Posts

MEAN Stack Interview Questions and Answers
Published On: June 19, 2024

C is the pioneering programming language that has both characteristics of assembly-level (low-level) and higher-level languages. It is also known as middle-level language and it is used to write an operating system and menu-driven consumer billing applications.

2. What are the features of C programming?

The following are the features of C Programming.

- Simple and Efficient
- Portable and Machine-Dependent
- Middle-level programming characteristics
- Structured programming language
- Function-rich libraries
- Dynamic memory management system
- Super-fast and extensible
- Pointers can be used in C.

3. Define a built-in function in C

The system provides built-in functions, or library functions, to enable developers to do common activities using pre-defined tasks.

For instance, If we want to print output into the terminal, we can use printf() in C.

The most popular built-in functions in C are scanf(), printf(), strcpy, strlwr, strcmp, strlen, strcat, and so on.

4. Define Preprocessor

A preprocessor is a software application that processes a source file before compilation. It includes header files, conditional compilation, line control, and macro expansion.

5. Define Pointer in C

A pointer is a variable that stores the address of another variable. If the value of the variable is stored in a normal variable, then the address of the variable is stored in a pointer variable.

Introduction Since MEAN Stack combines several other applications as part of its functionality, it is...

Top 15 Struts Interview Questions and Answers
Published On: June 18, 2024

Struts Interview Questions and Answers When it comes to developing Java web applications, Struts is...

Top 20 C Sharp Interview Questions and Answers
Published On: June 17, 2024

C Sharp Interview Questions and Answers Microsoft created the general-purpose programming language C# together with...

Top 20 VB.Net Interview Questions and Answers
Published On: June 17, 2024

VB.Net Interview Questions and Answers A wide range of applications, including desktop, web, and mobile...

6. Explain the various types of decision control statements

The control statements are used to execute or convert the control from one portion of the program to another following the condition. The statements written in a program are implemented sequentially from top to bottom.

- If – else statement
- Normal if-else statement
- Else-if statement
- Nested if-else statement
- Switch statement

7. What are the real-world applications that can be developed using C++?

C++ is the powerful programming language used to develop the following kinds of applications

- Operating System
- GUI Based Applications
- Distributed Systems
- Database Software
- Banking Applications
- Advanced Computations and Graphics
- Embedded Systems

8. What are the various data types used in C++?

There are 4 major data types used in C++ and they are

- The primitive data type is a basic datatype like char, short, int, long, float, bool, or double.
- Derived data types include arrays, pointers, and so on.
- Enumeration data types like Enum
- User-defined data types, like structure or class

C++ developer Salary

C and C++ Interview Questions and Answers for Experienced

9. Define recursion in C

Recursion is the thing that happens when a function in C calls a copy of itself. In simple terms, when a function calls itself, it is known as recursion and the function is known as a recursive function.

Syntax

```
void do_recursion()

{

… .. …

do_recursion();

… .. …]

}

int main()

{

… .. …

do_recursion();

… .. …

}
```

10. Differentiate global int and static int declaration with example

The scope is only the difference between static and global integers. The global_temp is a global variable that is known to everything inside the program and it will be visible when we add “extern int global_temp;” in the source file.

```
#include <stdio.h>
```

```
int my_global_var = 0;
int
main(void)
{
  printf(“%d\n”, my_global_var);
  return 0;
}
```

A static variable has a local scope that doesn’t allocate the variable in the stack segment of the memory. It will have less than global scope and it resides in the .bss segment of the combined binary.

```
#include <stdio.h>
int
myfunc(int val)
{
   static int my_static_var = 0;
   my_static_var += val;
   return my_static_var;
}
int
main(void)
{
  int myval;
  myval = myfunc(1);
  printf(“first call %d\n”, myval);
  myval = myfunc(10);
```

```
   printf("second call %d\n", myval);

   return 0;

}
```

11. Define typecasting in C

Typecasting is the process that converts a variable from one data type to another and if we want to store the large type value to an int type, then we can convert the data type into another data type explicitly.

Syntax

```
int x;

for(x = 12; x<=54; x++)

{

printf("%c", (char) x); /*Explicit casting from int to char*/

}
```

12. Explain dangling pointers

The dangling pointer points to a memory that has been freed already and the storage will no longer be allocated. When we try to access it, that might lead to a segmentation fault. To end this, we can use a dangling pointer.

```
#include<stdio.h>

#include<string.h>

char *func()

{

    char str[10];

    strcpy(str,"Hello!");
```

```
 return(str);

}
```

Here, the local variable will return an address that can be scoped by the time control that was returned to the calling function.

```
*c = malloc (sizeof(int));

free(c);

*c = 3;
```

13. Write a program to remove duplicates in an array

```
#include <stdio.h>

int main()

{

   int n, a[100], b[100], calc = 0, i, j,count;

   printf(“Enter no. of elements in array.n”);

   scanf(“%d”, &n);

   printf(“Enter %d integers”, n);

   for (i = 0; i < n; i++)

       scanf(“%d”, &a[i]);

       for (i = 0; i<n; i++)

       {

           for (j = 0; j<calc; j++)

           {

                if(a[i] == b[j])

                break;

            }
```

```
          if (j== calc)

          {

                b[count] = a[i];

                calc++;

           }

     }

     printf(“Array obtained after removing duplicate
elements”);

     for (i = 0; i<calc; i++)

     {

            printf(“%dn”, b[i]);

     }

     return 0;

}
```

14. Write a program for checking whether the given number is palindrome or not without a recursive method

```
#include <stdio.h>

#include <conio.h>

int reverse(int num);

int isPalindrome(int num);

int main()

{

    int num;

    printf(“Enter a number: “);

    scanf(“%d”, &num);
```

```
if(isPalindrome(num) == 1)

{

 printf(“the given number is a palindrome”);

    }

    else

    {

        printf(“the given number is not a palindrome number”);

   }

    return 0;

}

int isPalindrome(int num)

{

    if(num == reverse(num))

    {

        return 1;

    }

    return 0;

}

int reverse(int num)

{

int rem;

    static int sum=0;

    if(num!=0){

            rem=num%10;
            sum=sum*10+rem;

            reverse(num/10);

    }

        else

             return sum;

    return sum;

}
```

Output

```
Enter a number: 1331

The given number is a palindrome
```

15. Write a program to find n’th Fibonacci Number

The fact that every number after the first two is the sum of the two numbers before it divides the Fibonacci number sequence.

Example, 0, 1, 1, 2, 3, 3, 8, 13, 21, 34, 55, 89, .. etc. Now, we have to find F{n} = F{n-1} + F{n-2} with base values F(0) = 0 and <code> F(1) = 1.

```
// Function to find the nth Fibonacci number

int fib(int n)

{

    if (n <= 1) {

        return n;

    }

    return fib(n – 1) + fib(n – 2);

}

int main()
```

```
{
   int n = 8;
   printf(“nth Fibonacci number is %d”, fib(8));
   return 0;
}
```

16. Explain class and objects in C++

A class is a user-defined data type that has data members and functions. Data variables are known as data members and functions are used to perform operations on the variables.

An object is an instance created for a class, as the class is a user-defined data type for an object to be called a variable of the data type.

Example

```
class A
{
private:
int data;
public:
void fun()
{
}
};
```

17. Differentiate struct and class in C++

The structure has members or variables that can be public by default but the members of the class are private by default. The default access specifier for a base class or struct is public when deriving a struct from a class but the default access specifiers will be private when deriving a class.

18. Explain polymorphism with its types and examples

Polymorphism is a thing that has many forms with various behaviors for different situations. It occurs when we use multiple classes for inheritance purposes. There are two types of polymorphism and they are compile-time polymorphism and runtime polymorphism.

Compile-time polymorphism: In this, we will know which method to call when the compile-time and the call are resolved by the compiler. It offers fast execution; which is known as compile time. It will be achieved by function overloading and operator overloading.

Example

```
int add(int a, int b){

      return a+b;

}

int add(int a, int b, int c){

      return a+b+c;

}

int main(){

    cout<<add(2,3)<<endl;

    cout<<add(2,3,4)<<endl;

    return 0;

}
```

Runtime Polymorphism: In this method, we will know which method to call during the runtime and the call will not be resolved by the compiler. It offers slow execution as it is known at runtime only. It will be achieved by virtual functions and pointers.

Example

```
class A
{
    public:
        virtual void fun()
{
            cout<<“base “;
         }
};
class B: public A
{
    public:
        void fun()
{
            cout<<“derived “;
         }
};
int main()
{
    A *a=new B;
    a->fun();
    return 0;
}
```

19. What are the access specifiers in C++?

C++ programming language has the following access specifiers

Public: All the data members and member functions can be accessed outside the class

Protected: All the data members and member functions are accessible inside the class and the derived class.

Private: No data members or member functions can be called outside the class

20. How to call a virtual function from a constructor

We can call a virtual function from a constructor but the behavior is a little varied. Once the virtual function is called, it resolves a virtual call at runtime. It is the number function of the present class that gets called and the virtual machine will not work within the constructor.

Example

```
class base
{
private:
   int value;
  public:
   base(int x){
    value=x;
   }
   virtual void fun()
{
   }
}

class derived

{

private:

  int a;

 public:

  derived(int x, int y):base(x){

   base *b;

   b=this;

   b->fun();      //calls derived::fun()

  }

  void fun(){

   cout<<”fun inside derived class”<<endl;

  }

}
```

C C++ Training

Conclusion

C and C++ are the in-demand skills for every IT aspirant to understand the fundamentals of programming. We provide updated **C and C++ interview questions and answers** to help clear the technical rounds easily. Learn the best **C and C++ training in Chennai.**

Share on your Social Media

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai – 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Navigation

- About Us
- Blog Posts
- Careers
- Contact
- Placement Training
- Corporate Training
- Hire With Us
- Job Seekers
- SLA’s Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Courses

- Python
- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS
- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking

Social Media Links

Review Sources

- Google
- Trustpilot
- Glassdoor
- Mouthshut
- Sulekha
- Justdial

VBA Macros

Mobile App Development

DevOps

Ambitionbox

Indeed

Software Suggest

Sitejabber