Share on your Social Media

C and C++ Interview Questions and Answers

Published On: January 24, 2024

C and C++ Interview Questions and Answers

The procedural programming language C was one of the first computer languages. Because of its structure, high-level abstraction, machine-independent features, etc., it is one of the most widely used programming languages and a fantastic place for beginners to start learning the language. On the other hand, Bjarne Stroustrup at Bell Labs created C++, a potent and versatile programming tool. This is by far one of the quickest object-oriented programming languages; it is an extension of C. Because of its great speed and interoperability, C++ is very popular. Explore a wide range of **IT courses** to gain expertise in the C and C++ programming languages.

1. C programming: what is it?

One of the earliest programming languages is C, which Dennis Ritchie created in 1972. It is a popular high-level, general-purpose, structured programming language used for a variety of activities, including creating operating systems, desktop programs, system apps, and Internet of Things apps. A large portion of Windows, UNIX, and Linux operating systems employ this straightforward and adaptable language for their scripting system

Related Courses

Related Posts

Manual Testing Interview Questions and Answers

Published On: February 21, 2024

Manual Testing Interview Questions and Answers Before a software application is released into production, manual…

applications.

2. What makes C a programming language categorized as mid-level?

C possesses features of both lower-level and higher-level, or assembly-level, languages. C is hence frequently referred to as a middle-level language. It is possible to develop an operating system and a menu-driven consumer billing system in C.

Suggested Article: A quick start to C Programming

3. Why is C referred to as a mother language?

Indeed. Because the majority of JVMs, compilers, and kernels are written in C, it is referred to as the mother tongue. You can pick up other programming languages quickly if you know C.

4. Who created the C programming language? The C language was created when?

In 1972, Dennis M. Ritchie created C at AT&T Bell Laboratories.

5. In C, what is a token?

Tokens are the discrete components of a program. The six types of tokens are accessible in C, and they are identifiers, keywords, constants, operators, special characters, and strings.

6. Which C storage class specifiers are there?

Any variable can be stored using storage classes. In C, there are four types of storage auto, register, extern, and static.

7. Which data types does the C programming language support?

This is a typical interview question for C programming.

Amazon Fresher Salary

Published On: February 21, 2024

Amazon Fresher Salary The exceptional chance to shape your professional path by having total control…

Components of Selenium

Published On: February 21, 2024

Components of Selenium Introduction In the field of software testing and automation, selenium is the…

Differences between SQL and PLSQL

Published On: February 21, 2024

Differences between SQL and

The data type defines programming data types. There are several specified data types in the C language with varying storage capacities:

Built-in data types: char, double, float, void, and int.

Derived data types: references, pointers, and arrays.

User-defined data types: Union, arrangement, and Listing

8. What is the pointer in C?

In C, variables called pointers are used to store the address of other variables. The memory for the variable is allocated during runtime. The pointer may represent a range of data types, such as char, float, double, and int.

9. What does a function in C do?

Any computer language's fundamental building blocks are its functions. The function is used in the writing of all C programs to preserve readability and reusability.

Applications of functions in C:

- By calling functions as needed, programs can reuse them repeatedly.
- Functions are used in C to prevent code rewriting.
- Any C program that is separated into functions is simple to follow.

In C programming, how many different kinds of operators are there?

A symbol called an operator is used to manipulate the values of variables. In C programming, a variety of operators are utilized, such as arithmetic operators, relational operators, logical operators, assignment operators, increment and decrement operators, bitwise operators, conditional operators, and special operators.

PLSQL Introduction SQL is the language that is predominantly used for…

10. What is dynamic memory allocation in C?

Allocating memory during the running time is called dynamic memory allocation. The C language has a set of functions called calloc(), malloc(), realloc(), and free() that are used for dynamic memory management.

11. What is C++?

Bjarne Stroustrup developed the object-oriented programming language C++. In 1985, it was made available. With the substantial addition of C language classes, C++ is a superset of C. Stroustrup first named the new language “C with classes.”. But the name was changed to C++. The C increment operator ++ is where the concept of C++ originated.

Explore **Object-Oriented Programming concepts in Java** and understand the basics of C++

12. What benefits does C++ offer?

In addition to preserving every characteristic of the C language, C++ adds other capabilities and makes memory management easier, such as

- Because C++ is a highly portable language, applications written in it can operate on any platform.
- Classes, objects, inheritance, polymorphism, and abstraction are all included in the object-oriented programming language C++.
- The inheritance notion is present in C++.
- Reusing pre-existing classes and getting rid of unnecessary code are both possible via inheritance.
- Data concealment enables programmers to create safe programs that are impervious to outside attacks.
- The function library in C++ is extensive.

Learn more about **polymorphism in OOPs**

13. What are the primary differences between C and C++?

- While C doesn't support references, C++ does.
- C++ comes with built-in features including templates, virtual functions, function overloading, friend functions, and inheritance. The C programming language does not include these.
- The classic if-else technique of exception handling is used in C. However, exception management is supported at the language level in C++.
- In C, scanf() and printf() are frequently used for input and output, respectively. The standard input stream in C++ is called cin, and the standard output stream is called cout.
- Although C++ supports both procedural and object-oriented programming techniques, C is a procedural programming language.

Get started with an understanding of **OOPs concepts in Python**

14. In C++, what is a namespace?

How will the compiler know which function to use if there are two or more functions with the same name defined in separate libraries? Namespace so enter the picture. A namespace establishes a scope and distinguishes variables, classes, functions, and other items with the same name that are present in several libraries. "Namespace" is the first keyword in the namespace.

Syntax:

namespace namespace_name {

// code declarations

}

15. In C++, what is a template?

In C++, data types are passed as parameters via a template. These simplify and ease the usage of classes and functions.

template(class type)ret-type func-name(parameter list){

//body of the function

}

16. In C++, what is a pointer?

In C++, pointers are a form of data that is used to hold another variable's memory address.

Pointer Declaration Syntax: *variable_name datatype;

Pointer Declaration Syntax

*datatype *variable_name;*

Suggested Read: Java vs. Python: Which is best for you?

17. In C++, what is a function?

In C++, a function is a collection of statements that work together to accomplish a certain goal. The operating system calls the main function to execute our code.

There is always at least one function in a C/C++ program with the name main.

Syntax:

Return_type Function_name (Parameters)

{

// Function Body

}

In C++, what is a destructor?

In C++, destructors are unique functions or procedures that eliminate an object's memory allocation. Usually, they get called when an object's scope is about to expire. For example, you can call a destructor when a function terminates.

They share the same name as the class – syntax – ~ (classname)();

18. What does the C++ language's void main() function accomplish?

There are two processes involved in running a C++ application. The first is compilation, which is the process of converting C++ code to object code. The second stage, linking, involves integrating the programmer's object code with that of libraries. The C++ language's main() method controls this function.

19. What does an inline function mean?

When a function is inline, the compiler copies its code to each compile-time location where the function is invoked. The fact that an inline function does away with the overhead of typical function calls is one of its main benefits. Explore **Google salaries for freshers**.

20. A scope resolution operator: what is it?

The following represents a scope resolution operator:

- This operator links a function declaration to a certain class.
- The following uses are made of the scope operator:
- When you have a local variable with the same name, you can use it to access a global variable.
- To define a function that is not class-specific.

21. How many different methods are there to use a constant to initialize an integer?

There are two methods:

- Conventional C notation is used in the first format – int result = 10;
- Constructor notation is used in the second format – int result (10);

22. What distinguishes the assignment operator (=) from equal to (==)?

Equal to (==) and the assignment operator (=) are two entirely distinct operators in C++. The equality relational operator equal to (==) determines whether two expressions are equal and returns true if they are and false otherwise. A variable can have a value assigned to it using the assignment operator (=). Therefore, under the equality relational operator for evaluation, we can have a complex assignment operation.

23. What makes delete different from delete[]?

To release memory allocated to an array that was allocated using new[], use the “delete[]” function. To free up a single memory chunk that was allocated using new, use the “delete” command.

24. Describe C++ storage qualifiers.

Const: This variable indicates that a program should not change memory once it has been initialized.

Volatile: This variable indicates that even though nothing in the program code changes the contents, the value in the memory location may change.

Mutable: This variable indicates that even while a specific structure variable, class, or class member function is constant, a specific member of a structure or class may be changed.

25. Describe the hanging pointer.

A dangling pointer is created when an object’s address is used after its lifetime has ended.

These scenarios include, for instance, utilizing the address of the memory block after it has been released or having a function return the addresses of the automatic variables.

26. What access specifiers are there in C++?

The following access specifiers are available in C++:

Public: Every member function and data member is reachable from outside the class.

Protected: The derived class and the class itself can access all data members and member functions.

Private: No member function or data member is available to users outside of the class.

Useful Source: Java Programs for Practice

Advanced C and C++ Interview Questions for Experienced

27. Explain the purpose of the #line in C.

When writing code that accepts a parameter as its line number, the preprocessor #line in C is used to reset the line number. Here’s an example of the same.

```
#include <stdio.h> /*line 1*/

/*line 2*/

int main(){ /*line 3*/ /*line 4*/

printf(“Hello worldn”); /*line 5*/

//print current line /*line 6*/

printf(“Line: %dn”,__LINE__); /*line 7*/
//reset the line number by 36 /*line 8*/

#line 36 /*reseting*/

//print current line /*line 36*/

printf(“Line: %dn”,__LINE__); /*line 37*/

printf(“Bye bye!!!n”); /*line 39*/

/*line 40*/

return 0; /*line 41*/

} /*line 42*/
```

28. How can one change a number into a string?

The function requires the writing of a format string as a string in a buffer and accepts a pointer to an array of char elements that need to be transformed.

```
int sprintf(char *str, const char *format, …)

#include<stdio.h>

#include <math.h>

int main()

{

char str[80];

sprintf(str, “The value of PI = %f”, M_PI);

puts(str);

return 0;

}
```

Output: Value of Pi = 3.141593

Discover the various **types of Java applications**

29. In C, what is recursion?

Recursion in C refers to the process by which a function calls a copy of itself. Put another way, this method is known as recursion and occurs when a function calls itself. The term “recursive function” is also applied to this function.

Syntax of Recursive Function:

```
void do_recursion()

{

… .. …

do_recursion();

… .. …]

}

int main()

{

… .. …

do_recursion();

… .. …

}
```

30. What differentiates a static int declaration from a global int declaration?

There is a scope difference between these. A global variable is observable throughout your application and has a global reach.

```
#include
```

```
int my_global_var = 0;
int
main(void)
{
  printf("%dn", my_global_var);
  return 0;
}
```

Global_temp is a global variable that is accessible to all programs; however, if your project consists of multiple files, you will need to add an “extern int global_temp;” to other source files to make it available in other modules.

Although the variables of a static variable are not allocated in the memory’s stack segment, they do have a local scope. Similar to global variables, it can have a scope that is smaller than global, but it still lives in the compiled binary ‘.bss segment’.

```
#include
 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 %dn”, myval);

  myval = myfunc(10);

  printf(“second call %dn”, myval);

  return 0;

}
```

Explore a wide range of **courses after 12th** grade and get started with your desired domain.

31. What differentiates a C++ struct from a union?

A struct is a collection of intricate data structures kept in a memory block, each of which has it’s memory location so that it may be accessed immediately.

In contrast, when one member variable in the union is assigned a value, it affects all other members since all of the member variables are kept in the same place in memory.

```
/* struct and union definitions*/

struct bar {

int a;

char b;

} bar;
```

```
union foo {
int a;
char b;
} foo;
/* using struc and union variables*/
struct bar y;
y.a = 3;
y.b = ‘c’;
union foo x;
x.a = 3;
x.b = ‘c’; // No, this affects the value of x.a!
```

Bring the output for the following code in C++?

```
#include <iostream>
class A {
public:
  A() {}
  ~A() {
    throw 42;
  }
};
int main(int argc, const char * argv[]) {
  try {
        A a;

        throw 32;

    } catch(int a) {

        std::cout << a;

    }

}
```

This software will end unexpectedly. Throw 32 will begin class A destruction and unwinding of the stack. During exception handling, the class A destructor will raise another exception, resulting in a program crash. This query checks the developer's familiarity with handling exceptions.

Resource to read: Latest IT Salary in India for Freshers

32. What is operator overloading?

To execute operations on user-defined data types, operator overloading is a crucial component. We can change the default meaning of operators like +, -, *, /, <=, etc. using operator overloading.

Example

The code below uses operator overloading to add two complex numbers:

```
class complex{

private:

 float r, i;

public:

 complex(float r, float i){
  this->r=r;
  this->i=i;
 }
 complex(){}
 void displaydata(){
  cout<<”real part = “<<r<<endl;
  cout<<”imaginary part = “<<i<<endl;
 }
 complex operator+(complex c){
  return complex(r+c.r, i+c.i);
 }
};
int main(){
complex a(2,3);
complex b(3,4);
complex c=a+b;
c.displaydata();
return 0;
}
```

33. Describe the C++ constructor.

A member function called the constructor is launched automatically each time an object is formed. For the compiler to recognize that a member function is a constructor, constructors share the same name as the class of which they are members. Additionally, constructors don’t use a return type.

Example:

```
class A{
 private:
  int val;
 public:
  A(int x){          //one argument constructor
   val=x;
  }
  A(){               //zero argument constructor
  }
}
int main(){
 A a(3);
 return 0;
}
```

Useful Resource: Method Overloading in Python

34. How much do you know about the friend function and friend class?

A friend class can access members of other classes that it has designated as friends, whether they are private, protected, or public.

Similar to the friend class, the friend function has access to members who are public, private, and protected.

Member functions are not friend functions, though.

Example

```
class A{
 private:
  int data_a;
 public:
  A(int x){
   data_a=x;
  }
  friend int fun(A, B);
}
class B{
 private:
  int data_b;
 public:
  A(int x){
   data_b=x;
  }
  friend int fun(A, B);
}
int fun(A a, B b){
 return a.data_a+b.data_b;
}

int main(){

 A a(10);

 B b(20);

 cout<<fun(a,b)<<endl;

 return 0;

}
```

The **top 15 interesting facts about Java** may amaze you.

35. What is a constructor of copies?

A member function known as a copy constructor uses an object of the same class to initialize another object.

Example

```
class A{

int x,y;

A(int x, int y){

 this->x=x;

 this->y=y;

}

};

int main(){

A a1(2,3);

A a2=a1;     //default copy constructor is called

return 0;
}
```

36. What differentiates pure virtual functions from virtual functions?

A member function from the base class that you redefine in a derived class is called a virtual function. Using the virtual keyword, it is declared.

Example

```
class base{
public:
 virtual void fun(){
 }
};
```

When 0 is assigned, a pure virtual function is declared; it is not implemented. There is no body there.

Example

```
class base{
public:
 virtual void fun()=0;
};
```

In this case, the value 0 is not assigned to anything, and the = sign has no bearing on the assignment. Its purpose is to inform the compiler that a function will be pure—that is, it won't have any users. Learn about the **importance and benefits of Android**.

37. Can a constructor be used to invoke a virtual function?

It is possible to invoke a virtual function from a constructor, yes. However, in this instance, the behavior is a little different. A virtual call is handled at runtime when a virtual function is called. The current class's member function is always called first. That is, the constructor does not function with the virtual machine.

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;

  }

}
```

In C++, how is memory allocated and deallocated?

In C++, memory deallocation is handled by the deletes operator, while memory allocation is handled by the new operator.

Example:

```
int value=new int;  //allocates memory for storing 1 integer

delete value;          // deallocates memory taken by value

int *arr=new int[10];    //allocates memory for storing 10 int

delete []arr;            // deallocates memory occupied by arr
```

Frequently Asked Programs for C and C++ Interviews

Find which of the three numbers is the largest.

```
#include <stdio.h>

int main()

{
    int a = 1, b = 2, c = 3;
    if (a > b && a > c)
        printf(“%d”, a);
    else if (b > a && b > c)
        printf(“%d”, b);
    else
        printf(“%d”, c);
    return 0;
}
```

Output

```
3
```

Also Read: **Infosys Salary for Freshers**

Write a program to find out if a given integer is prime or not

```
#include <stdio.h>
int main()
{
    int N = 91;
    int flag = 0;
    for (int i = 2; i <= N / 2; i++) {
        if (N % i == 0) {
            flag = 1;
            break;
        }
    }
    if (flag == 0)
        printf(“Not a Prime Number”);
    else
        printf(“Is a Prime Number”);
    return 0;
}
```

Output

```
Is a prime number.
```

Write a C program to find compound interest.

```
#include <stdio.h>
#include <math.h>
int main()
{
    double principal = 2300;
    double rate = 7;
    double time = 4;
    double amount
        = principal * ((pow((1 + rate / 100), time)));
    double CI = amount – principal;
    printf(“Compound Interest is : %lf”, CI);
    return 0;
}
```

Output

```
The compound interest is: 714.830823
```

Explore the **top 10 software courses for high-paying jobs**

To convert a binary number to a decimal value, write a program.

```
#include <stdio.h>
int main()
{
    int N = 11011;
     int a = 1;
    int ans = 0;
    while (N != 0) {
        ans = ans + (N % 10) * a;
        N = N / 10;
        a = a * 2;
    }
    printf(“%d”, ans);
    return 0;
}
```

Output

27

Write a program to check the Armstrong number

```
#include<stdio.h>
int power(int x, unsigned int y)
{
    if (y == 0)
        return 1;
    if (y % 2 == 0)
        return power(x, y / 2) * power(x, y / 2);
    return x * power(x, y / 2) * power(x, y / 2);
}
int order(int n)
{
    int res = 0;
    while (n) {
        res++;
        n = n / 10;
   }
    return res;
}
int isArmstrong(int x)
{
    int n = order(x);
    int temp = x, sum = 0;
    while (temp) {
        int r = temp % 10;
        sum += power(r, n);
        temp = temp / 10;
  }
    if (sum == x)
        return 1;
    else
        return 0;
}
int main()
{
    int x = 120;
    if (isArmstrong(x) == 1)
        printf(“Truen”);
    else
        printf(“Falsen”);
    x = 1634;
    if (isArmstrong(x) == 1)
        printf(“Truen”);
    else

        printf(“Falsen”);

    return 0;

}
```

Output

```
False

True
```

Propel your career by learning about **the future of software testing**.

Write a program to find the area of a circle.

```
#include <math.h>

#include <stdio.h>

#define PI 3.142

double findArea(int r) { return PI * pow(r, 2); }

int main()

{

   printf(“Area is %f”, findArea(5));

   return 0;

}
```

Output

```
Area is 78.550000
```

Write a program to create a pyramid pattern using C.

```
#include <stdio.h>
int main()
{
    int N = 5;
    for (int i = 1; i <= N; i++) {
        for (int j = 1; j <= N – i; j++)
            printf(” “);
        for (int j = 1; j < 2 * i; j++)
            printf(“*”);
        printf(“n”);
}
    return 0;
}
```

Output

```
    *
   ***
  *****
 *******
*********
```

To sort the first half in ascending order and the second half in descending order, write a program.

```
#include <studio.h>
void Sort_asc_desc(int arr[], int n)
```

```
{
    int temp;
    for (int i = 0; i < n – 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    for (int i = 0; i < n / 2; i++)
        printf(“%d “, arr[i]);
    for (int j = n – 1; j >= n / 2; j–)
        printf(“%d “, arr[j]);
}
int main()
{
    int arr[] = { 11, 23, 42, 16, 83, 73, 59 };
    int N = sizeof(arr) / sizeof(arr[0]);
    Sort_asc_desc(arr, N);
    return 0;
}
```

Output

```
11 16 23 83 73 59 42
```

Prepare for Python interviews by checking our **Python interview questions and answers**

Write a C program to sort arrays using bubble, selection, and insertion sort.

```
#include<studio.h>
void bubble_sort(int* arr, int n)
{
  for (int j = 0; j < n – 1; j++) {
    for (int i = 0; i < n – j – 1; i++) {
      if (arr[i] > arr[i + 1]) {
        int temp = arr[i];
        arr[i] = arr[i + 1];
        arr[i + 1] = temp;
      }
    }
  }
}
void swap(int* xp, int* yp)
{
  int temp = *xp;
```

```
    *xp = *yp;

    *yp = temp;

}

void selectionSort(int arr[], int n)

{

    for (int i = 0; i < n – 1; i++) {

        int min_idx = i;

        for (int j = i + 1; j < n; j++)

            if (arr[j] < arr[min_idx])

                min_idx = j;

        if (min_idx != i)

            swap(&arr[min_idx], &arr[i]);

    }

}

void insertionSort(int arr[], int n)

{

    for (int i = 1; i < n; i++) {

        int key = arr[i];

        int j = i – 1;

        while (j >= 0 && arr[j] > key) {

            arr[j + 1] = arr[j];

            j = j – 1;
        }
         arr[j + 1] = key;
    }
}
int main()
{
    int arr1[] = { 9, 4, 3, 11, 1, 5 };
    int arr2[] = { 4, 3, 9, 1, 5, 11 };
    int arr3[] = { 5, 1, 11, 3, 4, 9 };
    int n = 6;
    printf(“Non-Sorted array: “);
    for (int i = 0; i < n; i++)
        printf(“%d “, arr1[i]);
    printf(“n”);
    // sort array
    bubble_sort(arr1, n);
    // printing array
    printf(“Sorted array using Bubble sort: “);
    for (int i = 0; i < n; i++)
        printf(“%d “, arr1[i]);
    printf(“n”);
    printf(“Non-Sorted array: “);
```

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

    printf(“%d “, arr2[i]);

printf(“n”);

// sort array

insertionSort(arr2, n);

// printing array

printf(“Sorted array using Insertion Sort: “);

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

    printf(“%d “, arr2[i]);

printf(“n”);

printf(“Non-Sorted array: “);

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

    printf(“%d “, arr3[i]);

printf(“n”);

// sort array

selectionSort(arr3, n);

// printing array

printf(“Sorted array using Selection Sort: “);

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

    printf(“%d “, arr3[i]);

printf(“n”);

return 0;
```

```
}
```

Output

```
Non-Sorted array: 9 4 3 11 1 5
Sorted array using Bubble sort: 1 3 4 5 9 11
Non-Sorted array: 4 3 9 1 5 11
Sorted array using Insertion Sort: 1 3 4 5 9 11
Non-Sorted array: 5 1 11 3 4 9
Sorted array using Selection Sort: 1 3 4 5 9 11
```

Learn about **mobile app testing** along with best practices for testing mobile apps

What is a map and write a program to use the map in C++?

Maps are associative containers that hold a key value and a mapped value combination.

Syntax: map map_name;

```
#include <iostream>
#include <iterator>
#include <map>
using namespace std;
int main()
{
   map test;
   test.insert(pair(1, 2));
```

```
    test.insert(pair(2, 3));

    map::iterator itr;

    for (itr = test.begin(); itr != test.end(); ++itr) {

        cout << itr->first

        cout << itr->second << ‘n’;

    }

 return 0;    }
```

In C++, how can a string be reversed?

Here is an example of code that reverses a string.

```
#include<iostream>

#include<string.h>

using namespace std;

int main ()

{

    char n[50], t;

    int i, j;

    cout << “Enter a string: “;

    gets(n);

    i = strlen(n) – 1;

    for (j = 0; j < i; j++,i–)

    {

        t = s[j];
        s[j] = s[i];

        s[i] = t;

    }

    cout << “nReverse string : ” << s;

    return 0;

}
```

Conclusion

Softlogic Systems’ software courses are essential for success, whether you’re getting ready for your first job interview or trying to upskill in this always-changing tech industry. We offer premium information at reasonable costs, all to hasten your advancement within a specific time frame. Join the millions of people we’ve previously empowered; we can help you do the same. Check out the 100% hands-on **C and C++ training in Chennai**.

Share on your Social Media

Navigation

About Us

Blog

Contact Us

All Courses

EASY WAY TO IT JOB

SLA Institute

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

Trending Courses

- Tableau
- DotNet Development
- Software Testing
- Angularjs Programming
- MEAN Stack

Social Media Links