**C++ Interview Questions**

**Name:**

1. How do you decide which integer type to use?
2. What does extern mean in a function declaration?
3. What’s the auto keyword good for?
4. Define a linked list node which contains a pointer to itself.
5. How do I declare an array of N pointers to functions returning pointers to functions returning pointers to characters?
6. How can I declare a function that returns a pointer to a function of its own type?
7. What can I safely assume about the initial values of variables which are not explicitly initialized?
8. How can I create a two-dimensional dynamic array ?
9. How can I deallocate a two-dimensional dynamic array ?
10. What is a virtual function ?
11. How do I do dynamic binding ?
12. When should I use a local static variable in a function ?
13. When should I use “register” to declare a variable ?
14. Can I put a function into a struct ?
15. How do I initialize a pointer to a function?
16. Are arrays row-major or column-major ?
17. What is the difference between a normal pointer and a void pointer?
18. What will the following program do?  
    void main()  
    {  
    int i;  
    char a[]="String";  
    char \*p="New Sring";  
    char \*Temp;  
    Temp=a;  
    a=malloc(strlen(p) + 1);  
    strcpy(a,p); //Line number:9//  
    p = malloc(strlen(Temp) + 1);  
    strcpy(p,Temp);  
    printf("(%s, %s)",a,p);  
    free(p);  
    free(a);  
    } //Line number 15//  
        
    a) Swap contents of p & a and print:(New string, string)  
    b) Generate compilation error in line number 8  
    c) Generate compilation error in line number 5  
    d) Generate compilation error in line number 7  
    e) Generate compilation error in line number 1
19. enum number { a= -1, b= 4,c,d,e};  
    What is the value of e ?  
    (a) 7  
    (b) 4  
    (c) 5  
    (d) 15  
    (e) 3
20. What is the difference between overloading and overriding ?
21. What is the difference between private and protected access ?
22. When are copy constructors called?
23. What is difference between malloc()/free() and new/delete?
24. How do I overload an assignment operator ?
25. What is special to static data members and member functions in a class ?
26. what is the difference between "new" and "operator new" ?
27. What is the purpose of keyword “volatile” ?
28. What is the purpose of keyword “mutable” ?
29. What is pure virtual function? or what is abstract class?
30. Can a copy constructor accept an object of the same class as parameter, instead of reference of the object?
31. What is a local class? Why can it be useful?
32. What is a nested class? Why can it be useful?
33. How do you access the static member of a class?
34. What does extern "C" int func(int \*, Foo) accomplish?
35. Can you overload a function based only on whether a parameter is a value or a reference?
36. What happens when you make call "delete this;" ?
37. What is inline function?
38. Can you overload a function based only on whether a parameter is a value or a reference?
39. Can you think of a situation where your program would crash without reaching the breakpoint which you set at the beginning of main()?
40. How can I handle a constructor that fails?
41. How can I handle a destructor that fails?
42. What is the difference between const char \*myPointer and char \*const myPointer?
43. How are prefix and postfix versions of operator++() differentiated?
44. What is the difference between a pointer and a reference?
45. What is name mangling in C++??
46. What is "this" pointer?
47. What happens when you make call "delete this;" ?
48. What is Virtual Destructor?
49. What do you mean by Stack unwinding?
50. What is an Iterator class?
51. What is Memory alignment?
52. How do you write a function that can reverse a linked-list?
53. Tell how to check whether a linked list is circular.
54. What’s the output of the following program? Why?

#include <stdio.h>

main()

{

typedef union

{

int a;

char b[10];

float c;

}

Union;

Union x,y = {100};

x.a = 50;

strcpy(x.b,\"hello\");

x.c = 21.50;

printf(\"Union x : %d %s %f \n\",x.a,x.b,x.c );

printf(\"Union y :%d %s%f \n\",y.a,y.b,y.c);

}

1. Consider the following struct declarations:

struct A { A(){ cout << \"A\"; } };

struct B { B(){ cout << \"B\"; } };

struct C { C(){ cout << \"C\"; } };

struct D { D(){ cout << \"D\"; } };

struct E : D { E(){ cout << \"E\"; } };

struct F : A, B

{

C c;

D d;

E e;

F() : B(), A(),d(),c(),e() { cout << \"F\"; }

};

What constructors will be called when an instance of F is initialized? Produce the program output when this happens.

1. Anything wrong with this code?

T \*p = new T[10];

delete p;

1. Anything wrong with this code?

long value;

//some stuff

value &= 0xFFFF;

1. What does the following code do and why would anyone write something like that?

void send (int \*to, int \* from, int count)

{

int n = (count + 7) / 8;

switch ( count % 8)

{

case 0: do { \*to++ = \*from++;

case 7: \*to++ = \*from++;

case 6: \*to++ = \*from++;

case 5: \*to++ = \*from++;

case 4: \*to++ = \*from++;

case 3: \*to++ = \*from++;

case 2: \*to++ = \*from++;

case 1: \*to++ = \*from++;

} while ( --n > 0 );

}

}