Question: Define C++?

Answer: C++ is a computer programming language that is a superset of C wherein additional features are made in the C language.

Question: Can we call C++ OOPS? and Why?

Answer: Yes, C++ can be called OOPS. The full form of OOPS is an Object-Oriented Programming System, which means a paradigm that provides an application of various concepts, including data binding, polymorphism, inheritance, and various others.

Question: Define Class in C++?

Answer: Class is referred to as the designing of the user-defined data type. It reflects the different entities, attributes, and actions.

Question: Define Object in C++?

Answer: Object is an instance of the class. An object can have fields, methods, constructors, and related. For example, a bike in real life is an object, but it has various features such as brakes, color, size, design, and others, which are instances of its class.

Question: Define Encapsulation in C++?

Answer: Encapsulation is the process of binding together the data and functions in a class. It is applied to prevent direct access to the data for security reasons. The functions of a class are applied for this purpose. For example, the customers' net banking facility allows only the authorized person with the required login id and password to get access. That is too only for his/her part of the information in the bank data source.

Question: What is an abstraction in C++?

Answer: An abstraction in C++ is hiding the internal implementations and displaying only the required details. For example, when you send an important message through email, at that time, only writing and clicking the send option is used. This outcome is just the success message that is displayed to confirm that your email has been sent. However, the process followed in transferring the data through email is not displayed because it is of no use to you.

Question: **Briefly explain the concept of Inheritance in C++.**

Answer: C++ allows classes to inherit some of the commonly used state and behavior from other classes. This process is known as inheritance. **Question**: **Define access specifier and its various types in C++ Answer**: An access specifier offers how it is possible to define how the class members, i.e., functions and variables, will be accessed outside the class's scope. There are three types of access specifier in C++:

- Private – Such class members can't be accessed outside the class in which they are declared and are only accessible within the same class. Even child classes are disabled to access private members of its parent class.

- Protected – In addition to the class in which they are declared, the child classes can access its parent class's protected members.

- Public – Class members declared as public can be accessed throughout the program (code)

Question: Define a namespace?

Answer: A namespace is used for resolving the name conflict of the identifier, which is accomplished by placing them under various namespaces. This way, it helps in the logical division of the different codes.

Question: Define a class template?

Answer: A class template is a name given to the generic class. The use of the keyword template is made for defining a class template.

Question: What is the function of the keyword "Volatile"?

Answer: "Volatile" is a function that helps in declaring that the particular variable is volatile and thereby directs the compiler to change the variable externally- this way, the compiler optimization on the variable reference can be avoided.

Question: Define storage class in C++? Name some?

Answer: Storage class in C++ specifically resemble life or even the scope of symbols, including the variables, functions, etc. Some of the storage class names in C++ include mutable, auto, static, extern, register, etc.

Question: **Can we have a recursive inline function in C++?**

Answer: Even though it is possible to call an inline function from within itself in C++, the compiler may not generate the inline code. This is so because the compiler won’t determine the depth of the recursion at the compile time.

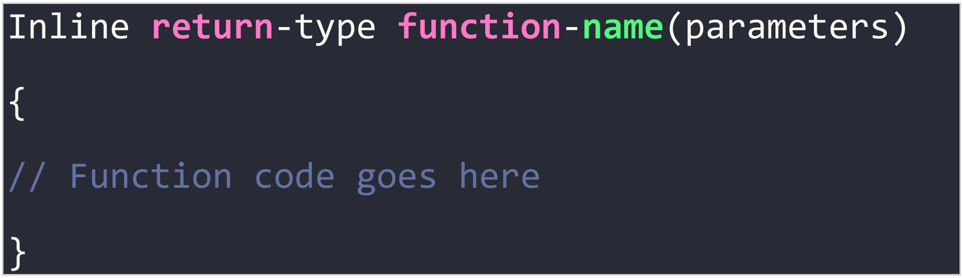
Nonetheless, a compiler with a good optimizer is able to inline recursive calls until some depth is fixed at compile-time and insert non-recursive calls at compile time for the cases when the actual depth exceeds run time.

Question: **Define an Inline Function in C++? Write its syntax. Is it possible for the C++ compiler to ignore inlining?**

Answer: In order to reduce the function call overhead, C++ offers inline functions. As the name suggests, an inline function is expanded in line when it is called.

As soon as the inline function is called, the whole code of the same gets either inserted or substituted at the particular point of the inline function call. The substitution is complete by the C++ compiler at compile time. Small inline functions might increase program efficiency.

The syntax of a typical inline function is:



```
Inline return-type function-name(parameters)

{

// Function code goes here

}
```

As the inlining is a request, not a command, the compiler can ignore it.

Question: **Explain ‘this’ pointer?**

Answer: The ‘this’ pointer is a constant pointer, and it holds the memory address of the current object. It passes as a hidden argument to all the nonstatic member function calls. Also, it is available as a local variable within the body of all the nonstatic functions.

As static member functions can be called even without any object, i.e., with the class name, the ‘this’ pointer is not available for them.

Question: **What are the most important differences between C and C++?**

Answer:

- C++ supports references while C doesn’t
- Features like friend functions, function overloading, inheritance, templates, and virtual functions are inherent to C++. These are not available in the C programming language.
- In C, exception handling is taken care of in the traditional if-else style. On the other hand, C++ offers support for exception handling at the language level.
- Mainly used input and output in C are scanf() and printf(), respectively. In C++, cin is the standard input stream while cout serves as the standard output stream.
- While C is a procedural programming language, C++ provides support for both procedural and object-oriented programming approaches.

Question: **Why do we need the Friend class and function?**

Answer: Sometimes, there is a need for allowing a particular class to access private or protected members of a class. The solution is a friend class, which can access the protected and private members of the class in which it is declared as a friend.

Similar to the friend class, a friend function is able to access private and protected class members. A friend function can either be a global function or a method of some class.

Some important points about friend class and friend function:

- Friendship is not inherited.
- Friendship isn’t mutual, i.e., if some class called Friend is a friend of some other class called NotAFriend, then it doesn’t automatically become a friend of the Friend class.
- The total number of friend classes and friend functions should be limited in a program as the overabundance of the same might lead to a depreciation of the concept of encapsulation of separate classes, which is an inherent and desirable quality of object-oriented programming.

Intermediate Level Interview Questions

Question: **Explain the significance of vTable and vptr in C++ and how the compiler deals with them**

Answer: vTable is a table containing function pointers. Every class has a vTable. vptr is a pointer to vTable. Each object has a vptr. In order to maintain and use vptr and vTable, the C++ compiler adds additional code at two places:

1. In every constructor – This code sets vptr:
    1. Of the object being created
    2. To point to vTable of the class
2. Code with the polymorphic functional call – At every location where a polymorphic call is made, the compiler inserts code in order to first look for vptr using the base class pointer or reference. The vTable of a derived class can be accessed once the vptr is successfully fetched. Address of derived class function show() is accessed and called using the vTable.

Question: **How is function overloading different from operator overloading?**

Answer: Function overloading allows two or more functions with different type and number of parameters to have the same name. On the other hand, operator overloading allows for redefining the way an operator works for user-defined types.

Question: **Is it possible for a C++ program to be compiled without the main() function?**

Answer: Yes, it is possible. However, as the main() function is essential for the execution of the program, the program will stop after compiling and will not execute.

Question: What is a destructor?

Answer: A destructor is the member function of the class. It has the same name as the class name and also prefixed with a tilde symbol. It can be executed automatically whenever an object loses its scope.

Question: Can we overload a destructor?

Answer: No, a destructor cannot be overloaded, and it has the only form without the parameters.

Question: What is the default constructor?

Answer: The compiler provides a constructor to every class in case the provider does not offer the same. This is when the programmer provides the constructor with no specific parameters than it is called a default constructor. The code for default constructor can be displayed in the following example.

```
// Cpp program to illustrate the
// concept of
Constructors #include
<iostream> using
namespace std; class
construct { public:
int a, b;
    // Default Constructor
construct()
    {
a = 10;
b = 20;
    } };
int main()
{
    // Default constructor called automatically
    // when the object is created
construct c;
    cout << "a: " << c.a << endl
<< "b: " << c.b;     return 1;
}
```

Question: Can we provide one default constructor for our class?

Answer: No, we cannot provide one default constructor for our class. When a variable in the class type is set to null, it means that it was never initialized and the outcomes will be zero.

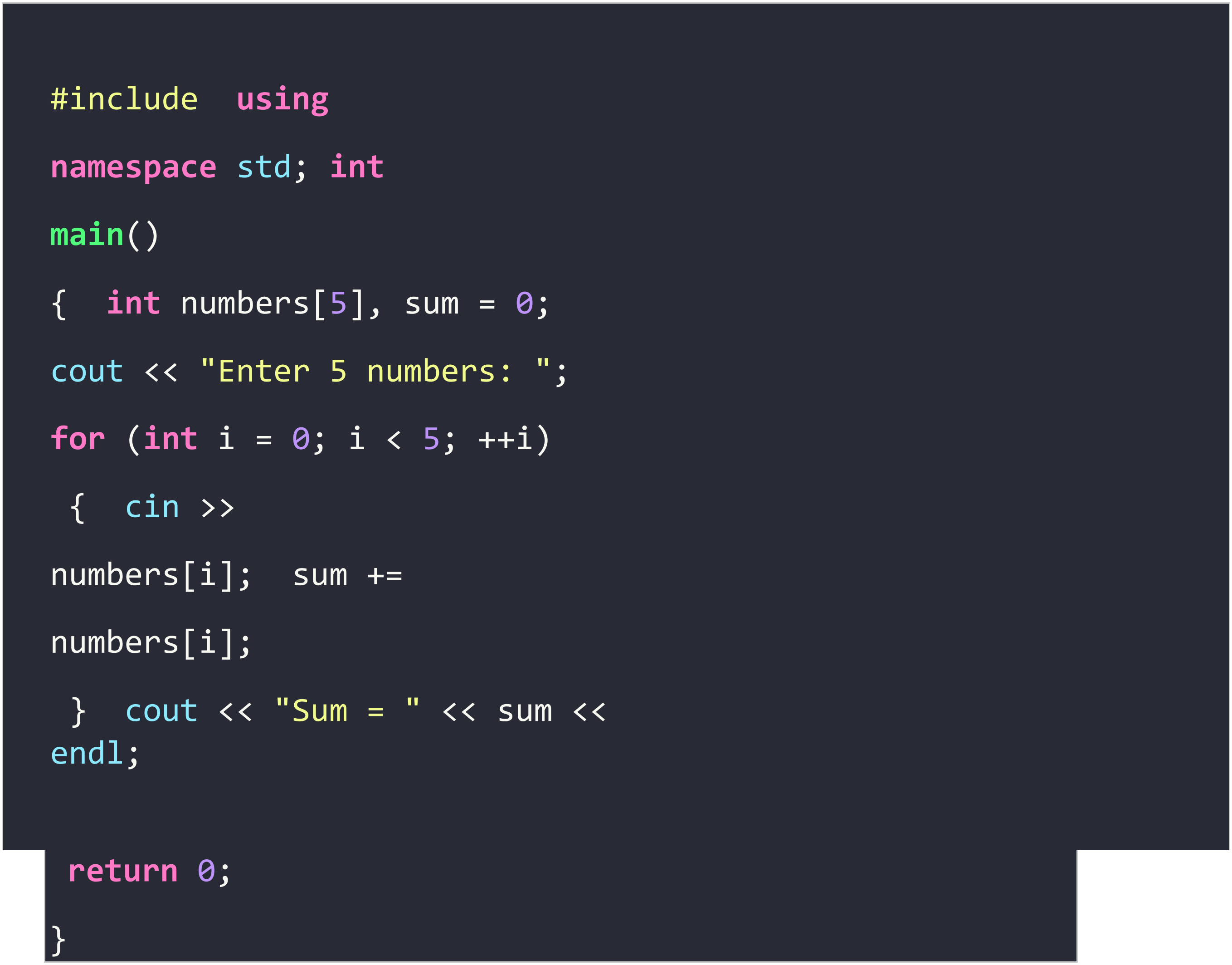
Question: What is the main difference between the keyword struct and class?

Answer: The keyword struct is used for resembling public members by default, while the keyword class is used for resembling private members by default.

Question: **Draw a comparison between C++ and Java Answer**:

- C++ has destructors, which are invoked automatically when an object is destroyed. Java has something called automatic garbage collection
- C++ supports multiple inheritance, operator overloading, pointers, structures, templates, and unions. Java doesn’t have any of them
- Java has a Thread class that is inherited in order to create a new thread. C++ has no inbuilt support for threads
- In C++, a goto statement offers a way to jump from a location to some labeled statement in the same function. There is no goto statement in Java
- C++ run and compile using the compiler, which converts the source code into machine level language. Hence, it is platform-dependent. Java compiler, on the other hand, converts the source code into JVM bytecode, which is platform-independent.

Question: **Take a look at the following C++ program:**



```
#include  using

namespace std; int

main()

{  int numbers[5], sum = 0;

cout << "Enter 5 numbers: ";

for (int i = 0; i < 5; ++i)

 {  cin >>

numbers[i];  sum +=

numbers[i];

 }  cout << "Sum = " << sum <<
endl;


 return 0;

}
```

Question: **Explain Virtual Functions and the concept of Runtime Polymorphism in C++ with a code example.**

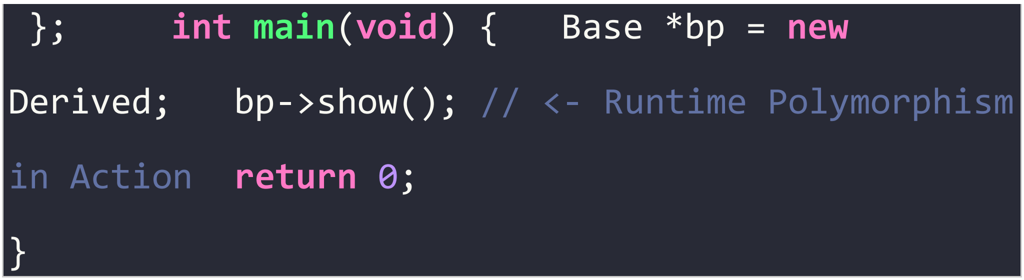
Answer: Any function when accompanying the virtual keyword exhibits the behavior of a virtual function. Unlike normal functions that are called in accordance with the type of pointer or reference used, virtual functions are called as per the type of the object pointed or referred.

In simple terms, virtual functions resolve at runtime, not anytime sooner. Use of virtual functions could also be understood as writing a C++ program leveraging the concept of runtime polymorphism. Things essential to writing a virtual function in C++ are:

- A base class
- A derived class
- A function with the same name in both the classes i.e. the base class and the derived class
- A pointer or reference of base class type that points or refers, respectively to an object of the derived class

An example demonstrating the use of virtual functions (or runtime polymorphism at play) is:

```
#include   using

namespace std;

class Base {

public:

 virtual void show() { cout<<" In Base \n"; }

 };  class Derived: public
Base {   public:


 void show() { cout<<"In Derived \n"; }
```



```
 };      int main(void) {    Base *bp = new
Derived;    bp->show(); // <- Runtime Polymorphism
in Action   return 0;
}
```

In the aforementioned program bp is a pointer of type Base. A call to bp>show() calls show() function of the Derived class. This is because bp points to an object of the Derived class.

Question: **What differences separate structure from a class in C++?**

Answer: There are two important distinctions between a class and a structure in C++. These are:

1. When deriving a structure from a class or some other structure, the default access specifier for the base class or structure is public. On the contrary, default access specifier is private when deriving a class.
2. While the members of a structure are public by default, the members of a class are private by default

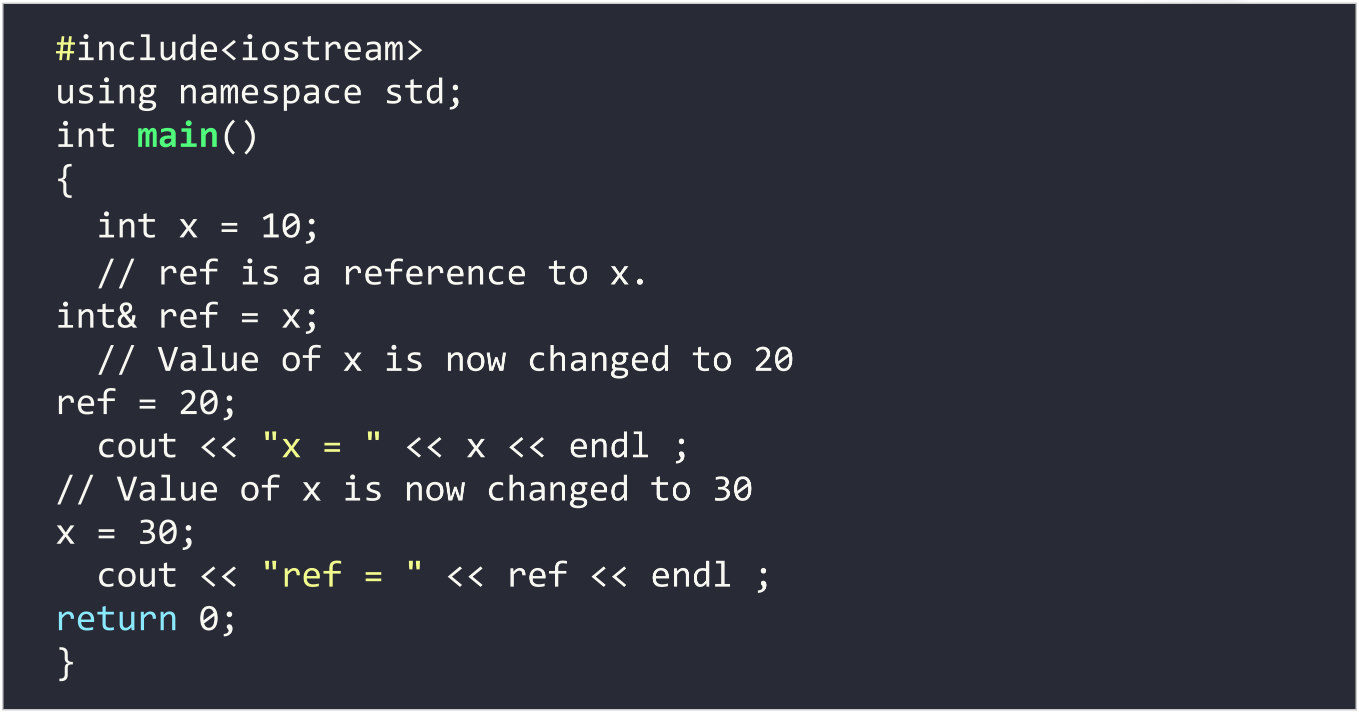
Question: **What does a Static member in C++ mean?**

Answer: Denoted by the static keyword, a static member is allocated storage, in the static storage area, only once during the program lifetime. Some important facts pertaining to the static members are:

- Any static member function can't be virtual
- Static member functions don't have 'this' pointer
- The const, const volatile, and volatile declaration aren't available for static member functions

Question: Define the Reference variable?

Answer: The reference variable in C++ is the name given to the existing variables. The variable name and reference variable point share the same memory location in C++, which helps in updating the original variable using the reference variable. The code can be displayed in the following example.



```
#include<iostream>
using namespace std;
int main()
{
  int x = 10;
  // ref is a reference to x.
int& ref = x;
  // Value of x is now changed to 20
ref = 20;
  cout << "x = " << x << endl ;
// Value of x is now changed to 30
x = 30;
  cout << "ref = " << ref << endl ;
return 0;
}
```

Advanced Level Interview Questions

Question: **Define the Copy Constructor used in C++ along with its general function prototype. Also, explain the various scenarios in which it is called.**

Answer: A member function that initializes an object using another object of the same class is known as a copy constructor in C++. Copy Constructor can also be made private. A call to the Copy Constructor can happen in any of the following 4 scenarios when:

1. The compiler generates a temporary object
2. An object is constructed or based on some another object of the same class
3. An object of the class is returned by value
4. An object of the class is passed (i.e., to a function) by value as an argument
5.

   Question: **Observe the following code snippet:**

After execution, what will be the value of I and j? Explain your answer.

Answer: Post the execution of the code above, i and j will be 6 and 5, respectively. For understanding the output, it’s important to understand how the unary ‘++’ operator and the decrement ‘--’ operator works in C++.

When any of these operators precede a variable, the value of the variable is first modified, and then this modified value is used. However, when any of the two operators follow a variable, the value is first used, and then it is modified.

Therefore, in the code above j is set to the unmodified value of 5, and then i is incremented to store 6.

Question: **Take a look at the following two code examples for printing a vector:**

```
Sample Code 1:

 vector vec; /* ... .. ... */ for (auto itr = vec.begin();

itr != vec.end(); itr++) {  itr->print();

}


Sample Code 2:

 vector vec; /* ... .. ... */ for (auto itr = vec.begin();

itr != vec.end(); ++itr) {  itr->print();

}
```

Is there any advantage of using one over the other?

Answer: Though both codes will generate the same output, sample code 2 is a more performant option. This is due to the fact that the post-increment 'itr++' operator is more expensive than the pre-increment '++itr' operator.

The post-increment operator generates a copy of the element before proceeding with incrementing the element and returning the copy. Moreover, most compilers will automatically optimize sample code 1 by converting it implicitly into sample code 2.

Question: **Suppose you have the GPA (Grade Point Average) of n number of students, and you need to store and display it using C++. Can you write a program that accomplishes this?**

Answer:

```
#include  #include

using namespace

std; int main()

{  int num;  cout << "Enter the total number of
students: ";  cin >> num;  float* ptr;  ptr =
new float[num];  cout << "Enter the GPA of
students." << endl;  for (int i = 0; i < num;
++i)



 {  cout << "Student" << i + 1 <<

": ";  cin >> *(ptr + i);

 }  cout << "\nDisplaying GPA of students." << endl;  for

(int i = 0; i < num; ++i) {  cout << "Student" << i + 1

<< " :" << *(ptr + i) << endl;

 }  delete []
ptr; return 0;
```

}

Question: What is a mutable storage class specifier? How can they be used?

Answer: A mutable storage class specifier is applied only on the class's non-static and non-constant member variable. It is used for altering the constant class object's member by declaring it. This can be done by using a storage class specifier.

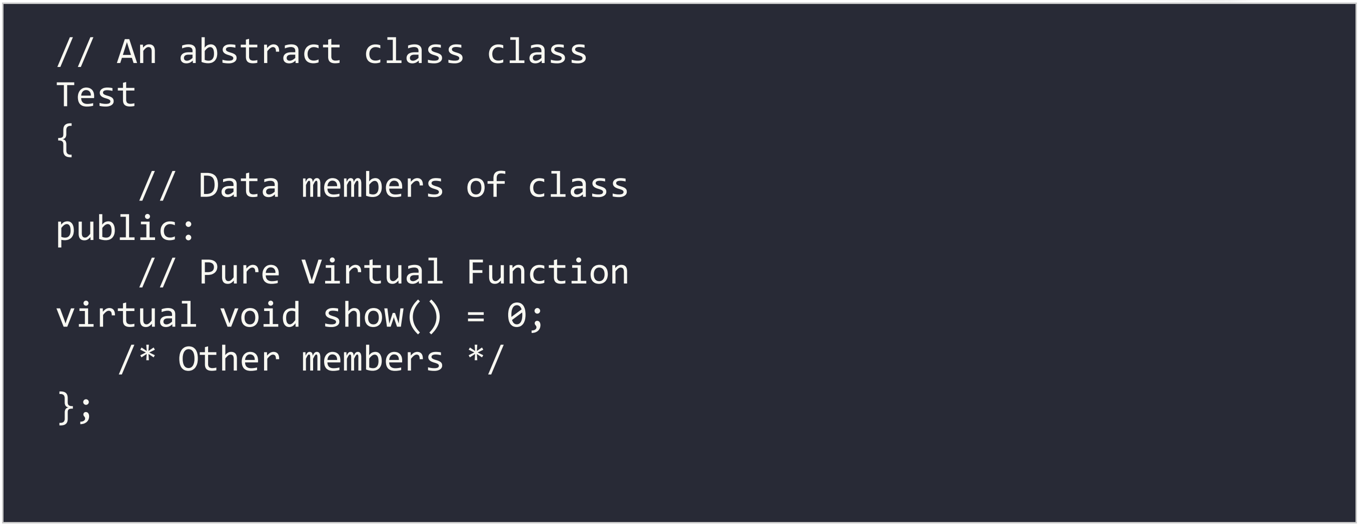
Question: What are the differences between a shallow copy and a deep copy?

Answer: The differences between a shallow copy and a deep copy can be stated as under.

Shallow Copy	Deep Copy
It allows memory dumping on a bit by bit basis from one object to another. It is achieved by using a copy instructor and an overloading	It allows the copy field, which is done by field from one object to another. It is used for shallow copy
assignment operator.	purposes.

Question: Define an Abstract class in C++?

Answer: An abstract class in C++ is referred to as the base class, which has at least one pure virtual function. In such a function, a person cannot instantiate an abstract class. This way, an Abstract class a pure virtual function is defined by using a pure specifier which is equal to zero during the declaration of the virtual member function in the class declaration. The code sample can be displayed as follows in example.



```
// An abstract class class
Test
{
    // Data members of class
public:
    // Pure Virtual Function
virtual void show() = 0;
   /* Other members */
};
```

Question; Can we have a String primitive data type in C++?

Answer: No, we cannot have a String Primitive data type in C++. Instead, we can have a class from the Standard Template Library (STL).

Question: Can we use access specifiers to achieve data hiding in C++?

Answer: Yes, we can use access specifiers to achieve data hiding in C++. These include Private and Protected.

Question: Define Block scope variable?

Answer: A Block scope variable is the one that is specified as a block using the C++ that can be declared anywhere within the block. **Question: What are the functions of the scope resolution operator?**

Answer: The functions of the scope resolution operator include the following.

1. It helps in resolving the scope of various global variables.
2. It helps in associating the function with the class when it is defined outside the class.

The code of the scope resolution operator can be displayed as follows.

```
#include <iostream>
using namespace
std;   int my_var =
0; int main(void) {
int my_var = 0;
   ::my_var = 1;  // set global my_var to 1
my_var = 2;    // set local my_var to 2
cout << ::my_var << ", " << my_var;
return 0;
}
```

Question: What is the function of the keyword "Auto"?

Answer: The keyword “Auto” is used by default for various local variables to make function work automatically.

Question: Define a token in C++? Give examples?

Answer: A token is a name given to the various functions in C++ programs. Examples of tokens include a keyword, symbol, string literal, identifier, constant, etc. The code of token in C++ other than C, can be displayed in the following example.

```
asm       bool     catch       class const_cast
delete dynamic_cast   explicit export     false
friend          inline mutable   namespace   new
operator  private       protected        public
reinterpret_cast  static_cast    template  this
throw true       try       typeid       typename
using     virtual  wchar_t
```

Question: **What is the ‘diamond problem’ that occurs with multiple inheritance in C++? Explain using an example.**

Answer: The diamond problem in C++ represents the inability of the programming language to support hybrid inheritance using multiple and hierarchical inheritance.

Suppose we have a university with some faculty members and some graduate students. A simple inheritance scheme in this scenario might have different types of people in different roles. However, all of them inherit from the same Person class.

The Person class defines an abstract getRole() method that would then be overridden by its subclasses in order to return the correct role type. Things up till this point is simple, however, if we wish to model the role of a TA or Teaching Assistant then things get weird.

A Teaching Assistant is both a student and a faculty member. This will yield the diamond problem, as illustrated in the figure below:

The problem generates an inheritance diagram resembling a diamond, hence the name, diamond problem.

Which getRole() implementation should the Teaching Assistant inherit? Graduate Student or the Faculty Member? A potential answer might be to have the Teaching Assistant class override the getRole() method and return a newly-defined role, say TA.

However, such an answer would also be far from complete as it will hide the fact that a teaching assistant is both a faculty member and a graduate student.

1. What is C++?

As an extension of the C language, C++ was developed by Bjarne Stroustrup as a general purpose cross-platform language which gives programmers a high level of control over system resources and memory.

2. What is namespace in C++?

If there are two or more functions with the same name defined in different libraries then how will the compiler know which one to refer to? Thus namespace came to picture. A namespace defines a scope and differentiates functions, classes, variables etc. with the same name available in different libraries. The namespace starts with the keyword “namespace”. The syntax for the same is as follows:

```
namespace namespace_name {
// code declarations }
```

3. How to input string in C++?

There are three ways to input a string, using cin, get, and getline. All three methods are mentioned in the sample program below.

```
#include <iostream>
using namespace std;

int main()
{
    char s[10];

    cout << "Enter a string: ";
cin >> str;

    cout << "\nEnter another string: ";
cin.get(s, 10);

    getline(cin, str);

return 0;
}
```

4. What is operator overloading in C++?

An overloaded declaration is a declaration in the same scope of function or operator declared with the same name more than once.

5. How to learn C++?

C++ is a programming language which is an extension of C. Thus, one should prefer to learn C first (it’s not necessary). After learning C, then understand the basic difference between C and C++. Implement all the basic programs you learnt in C in C++ also. Then dive into the OOPs concept of C++. Do as many hands-on as possible to understand basic OOPs, and then dive into advanced level OOPs. When all the basics are clear, build a small game to understand the structure and remain concepts if any. By following all these steps one can learn C++.

6. What is the difference between C and C++?

The difference between c and c++ is that C++ is a object oriented language, which means that it has all the features of C as well as its own thing that is the concept of OOP. C++ has many functionalities of OOP that are missing from C such as encapsulation, abstraction, classes, objects, etc.

7. How to reverse a string in C++?

To reverse a string, a sample code is mentioned below.

```
#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;
}
```

8. What is template in C++?

A template in C++ is used to pass data types as parameters . These make it easier and more simpler to use classes and functions.

```
template <typename T>
```

```
          int fun  (T a,T b)
          {
          return (a+b);
          }

int main(){
cout<<fun<int>(11,22);
}
```

9. What is using namespace std in C++?

Using namespace std in C++ tells the compiler that you will be making use of the name space called 'std'. The 'std' namespace contains all the features of the standard library. You need to put this statement at the start of all your C++ codes if you don't want to keep on writing std:: infront of every variable/string or whatever standard library feature you are making use of, as it becomes tedious to do so.

10. How to download turbo C++ for windows 10?

To download turbo c++ follow the steps mentioned below:
Step-1: Download turbo C++ from http://www.turboc8.com/p/download.html
Step-2: Extract Turbo.C.3.2.zip file.
Step-3: Run setup.exe file.
Step-4: Follow the instructions mentioned.

11. How to paste in turbo C++?

Paste in turbo C++ can be done by two techniques:

1. Shift+Insert
2. Open the file in notepad with .cpp extension. Make the changes and save it. After saving the file, you can open it from the Turbo C++ application file menu from where you stored the cpp file.

12. What is pointer in C++?

Pointers in C++ are a data type that store the memory address of another variable. For eg.

```
char *str = "Hi, How are you?";
                Here the pointer variable *str points to the
"Hi, How are you?"

                or

                int age;
                int *int_value;

                *int_value = &age;

                cout<<"Enter your age please:";
cin>>age;

                cout<<"\n Your age is:"<<*int_value;

                // this will print your age as the variable i
pointing to the variable age.
```

13. What is function in C++?

A function in C++ is a block of code that can be referenced from anywhere in the system and that serves a specific purpose.

```
                int fun(){
                int a = 11;
                return 11;
                }

        int main(){

               int b = fun();
}
```

14. What is destructor in C++?

Destructors in c++ are special function/methods that are used to remove memory allocation for objects. They are called usually when the scope of an object ends. eg. when a function ends you can call a destructor. They are of the same name as the class – syntax – ~<classname>();

15. Who invented C++?

Bjarne Stroustrup invented C++ in 1985.

16. How to convert integer to string in C++?

There are 2 approaches to convert integer variables to string. Both the approaches with a sample code are mentioned below.

```
Approach-1
#include<iostream>
#include<string>
using namespace std;
void main()
{      int n= 1;      string
s= to_string(n);      cout
<< s;
}

Approach-2

#include<iostream>
#include <sstream>
#include <string>
using namespace std;
int main()
{
    int n = 17;

    // declaring output string stream
ostringstream s1;

```

```
        // Sending a number as a stream
    into output str     s<< n;
        // the str() converts number into string
    string fin = s.str();      // Displaying the
    string     cout << fin;
      return 0;
    }
```

17. What is function overloading in C++?

Function Overloading happens in C++ when two or more functions share the same name. They can be differentiated on the basis of the type of data they are passing as parameters or even the number of paramters they are passing. eg. int fun(char a); & int fun(int b); & void fun(int a, int b)

18. What is stl in C++?

Stl is the standard template library. It is a library that allows you to use a standard set of templates for things such as: Algorithms, functions, Iterators in place of actual code.

```
queue<int> Q;

          for(k=0;k<10;k++)
          {
          Q.push(k);          }
```

19. How to run C++ program in cmd? verify gcc

```
    installtion using the command:

            $ gcc -v

    then go to your working directory or
    folder where your code is:        $ cd
    <folder_name>
```

then build the file containing your c code as such:

```
        $ gcc main.cpp

                or

        $ g++ -o main main.cpp
```

```
then run the executable generated in your
system:
        $ main.exe
```

20. What is type casting in C++?

Type casting in C is used to change the data type. They are of two types: Implicit Type Conversion: It is automatic. Explicit Type Conversion: It is user-defined.

21. How to use string in C++?

A string is a sequence of characters. In C++, string is a data type as well as a header file. This header file consists of powerful functions of string manipulation. A variable of string is declared as follows:

```
string str= "Hello";

And to use string one needs to include the
header file.


  // Include the string library
  #include <string>

// Create a string variable
string str= "Hello";
```

22. How to input string in C++ with spaces?

The code to input a string in C++ with spaces is as follows:

```
   #include <iostream>
```

```
#include <string>
using namespace std;

    int main()
    {
    string s;

    cout << "Enter the sentence";
    getline(cin, s);
    cout << str;
    return 0;  }
```

23. What is stream in C++?

Stream refers to a stream of characters to be transferred between program thread and i/o.

24. What is the difference between structure and class in C++?

The difference between structure and class is as follows:

- By default, the data members of class are private whereas data members of structure are public.
- While implementing inheritance, the access specifier for struct is public whereas for class its private.
- Structures do not have data hiding features whereas class does.
- Structures contain only data members whereas class contains data members as well as member functions.
- In structure, data members are not initialized with a value whereas in class, data members can be initialised.
- Structures are stored as stack in memory whereas class is stored as heap in memory.

25. How to clear screen in C++?

One can clear screen using – clrscr() or system(“clear”).

26. Who developed C++?

Bjarne Stroustrup in 1998 at Bell Labs developed the lanuage C++.

27. How to compile and run C program in notepad++ ?

To compile and run c program in notepad++ follow the steps mentioned below:
Step-1: Download and install notepad++
Step-2: Download and install MinGw gcc along with gcc.
Step-3: Configure notepad++ for gcc. This step can be further divided into two

sub-steps. A: Create C compiler tool in Notepad++ B: Creating C execution tool.

Step-4: Execute C program in Notepad++

28. How many keywords in C++ ?

There are 95 reserved keywords in C++ which are not available for re-definition or overloading.

29. What is iostream in C++?

It is a header file that includes basic objects such as cin, cout, cerr, clog.

30. How to give space in C++?

In C++ programming, the space can be given using the following code.
cout << ” ” ;

31. How to dynamically allocate a 2d array in C++ ?

There are several methods by which one can allocate memory to 2D array dynamically one of which is as follows.

```
#include <iostream>  int
main()
{
    int row = 2, col = 2;
    int* a =  new int[row * col];

    int i, j, count = 0;      for (i
= 0; i <  row; i++)          for (j =
0; j < col; j++)             *(a+
i*col + j) = count++;

    for (i = 0; i <  row; i++)
for (j = 0; j < col; j++)
         printf("%d ", *(a + i*col + j));
```

```
        delete[ ] a;
    return 0;
18 }
```

32. How to use goto statement in C++ ?

Goto statement provided unconditional jump in the code. The syntax is: goto label;

```
    label: statement;

    #include <iostream>
    using namespace std;

    void main () {
        float d, avg, add = 0.0;
    int j, n;
        cin >> n;

        for(j = 1; j <= n; ++j)
        {
            cout << "Enter number" << i;
    cin >> d;
            if(d < 0.0)
            {
                   goto jump;
            }
    add+= d;
        }     jump:
    avg = add/ (j- 1);
    cout << avg;
      }
```

33. What is function overriding in C++ ?

When a function with same name is present in both parent and child class then it is called function overriding.

```
#include <iostream>
using namespace std;
class parent { public:   void
display(){
cout<<"Parent Class";
   } };
class child: public parent{
public:    void display() {
cout<<"Child Class";
   } }; int main() {
child o = parent();
o.display();
   return 0;
}
```

34. Which operator cannot be overloaded in C++ ?

Some of the operators that cannot be overloaded are as follows:

– Dot operator- “.”
– Scope resolution operator- “::”
– “sizeof” operator
– Pointer to member operator- “.*”

35. How to copy and paste in turbo C++ ?

Press Ctrl + Insert to copy.
Press Shift + Insert to paste.

36. Why C++?

The use of C++ is varied such as:

– It is used in developing graphic user interface based applications like adobe photoshop.

– It is used in developing games as it overrides the complexity of 3D games. – There are many animated softwares developed in C++

– Most of the compilers are written in C++.

– Google Chrome, Mozilla Firefox etc. web browser are developed using C++

There are many more such uses that make C++ a desired language.

37. What is bool in C++?

Bool is a data type in C++ which takes two values- True and False. Syntax is as follows: bool b1 = true;

A sample code is as follows:

```
#include<iostream>
using namespace std;
int main()
{
    int a= 60, b= 70;
bool c, d;      c= a==
b; // false
           c= a< b;
// true
           cout
<<b1;      cout
<< b2 ;
    return 0;
}
```

38. What is exception in C++ ?

Runtime abnormal conditions that occur in the program are called exceptions. These are of 2 types:

– Synchronous

– Asynchronous

C++ has 3 specific keywords for handling these exceptions:

– try

– catch

– throw

39. How to set decimal places in C++ ?

For limiting the decimal places in C++ there are five functions : floor(), ceil(), trunc(), round() and setprecision(). Out of these five, only setprecision() function is used for setting the decimal places to put as output. All the functions are mentioned in the following sample code.

```
#include<bits/stdc++.h>  using
namespace std;

int main()
{
    float a =2.33333;     cout <<
floor(a) << endl;      cout <<
ceil(a) << endl;     cout <<
trunc(a) << endl;     cout <<
round(a) << endl;     cout <<
setprecision(2) << a;
    return 0;
}
```

40. How to get absolute value in C++?

In C++, there are three functions in the cstdlib header file to return the absolute value of the integer. Those are:

– abs()

– labs()

– llabs()

The syntax for all the functions is same – function_name(integer value)

The difference lies in the range for integer value being passed as an argument. For abs() its type int in C++. For labs(), its type long int in C++ and for llabs() its long long int in C++.

Sample code for the illustrating the three functions is as follows:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main()
{
    int a, b, c;

    a  =  abs(22);          b=
labs(1234355L);             c=
llabs(1234863551LL);
cout  <<  a;        cout  <<  b;
cout<< c;     return 0;
}
```

41. What is the difference between C++ and Java?

The difference between c++ and java are as follows:

– C++ supports goto statements whereas Java does not.
– C++ is majorly used in system programming whereas Java is majorly used in application programming.
– C++ supports multiple inheritance whereas Java does not support multiple inheritance
– C++ supports operator overloading whereas Java does not support operator overloading.
– C++ has pointers which can be used in the program whereas Java has pointers but internally.

– C++ uses a compiler only whereas Java uses both compiler and interpreter. –
C++ has both call by value and call by reference whereas Java supports only call by value.
– C++ supports structures and joins whereas Java does not support structure and joins
– Java supports unsigned right shift operator (>>>) whereas C++ does not. – C++
is interactive with hardware whereas Java is not that interactive with hardware.

42. How to concatenate string in C++ ?

The strings in C++ can be concatenated in two ways- one considering them string objects and second concatenating them C style strings.

```
#include <iostream>
using namespace std;

int main()
{
    string s_1, s_2, fin;
cout << "Enter string";
getline (cin, s_1);     cout
<< "Enter string ";
getline (cin, s_2);     fin=
s_1 + s_2;
    cout << fin;

    char str1[50], str2[50], fin[100];

    cout << "Enter string";
    cin.getline(str1, 50);

    cout << "Enter string";
    cin.getline(str2, 50);

    strcat(str1, str2);

    cout << "str1 = " << str1 << endl;
```

```
        cout << "str2 = " << str2;

        return 0;
    }
```

43. How to convert char to int in C++ ?

There are three methods for converting char variable to int type variable. These are as follows: – atoi()
– sscanf()
– typecasting

A sample code depicting all three functions are as follows:

```
    #include<stdio.h>
    #include<stdlib.h>
    int main() {    char
    *s = "6790";
    char d = 's';
       int a,b,c;

       sscanf(s, "%d", &amp;a); // Using sscanf
    printf("a : %d", a);

    b = atoi(s); // Using atoi()
       printf(“b : %d", b);

    c = (int)(d); // Using typecasting   printf("c
      : %d", c);

       return 0;
    }
```

44. How to generate random numbers in C++ with a range?

Using the rand() function we can generate random numbers in C++ within a range.

```
       #include <iostream>
```

```
#include <random>
int main()
{
int max=100, min=54,i;
int range = max - min + 1;
for (i=min; i<max;i++) {
     int num = rand() % range + min;
     cout<<num;
     }
     return 0; }
```

45. What is stack in C++?

A linear data structure which implements all the operations (push, pop) in LIFO (Last In First Out) order. Stack can be implemented using either arrays or linked list.The operations in Stack are – Push: adding element to stack
– Pop: removing element from stack
– isEmpty: returns true if stack is empty – Top: returns the top most element in stack

46. What is conio.h in C++?

Conio.h is a header file used for console input and output operations and is used for creating text based user interfaces.

47. How to find absolute value in C++?

To find the absolute value in c++, we can use abs() function. The abs() function in C++ returns the absolute value of an integer number.

```
#include <iostream> #include
<cstdlib>
using namespace std;

int main()
{
        int   a=3.456;
int   x    =    abs(a);
cout << x;
            return 0;
}
```

48. How to exit from turbo C++?

To exit Turbo C++, use the Quit option under the File Menu, or press Alt + X.

49. What is iterator in C++?

Any object which has an ability to iterate through elements of the range it has been pointing to is called iterator.

50. What is :: in C++?

:: is called a scope resolution operator which is used to access global variables with the same name as of local variables, for defining functions outside the class, for accessing static variables, and for referring to a class inside of another class.

51. What is enum in C++?

enum is abbreviation of Enumeration which assigns names to integer constant to make a program easy to read. Syntax for the same:

enum enum_name{const1, const2, ……. };

52. What is endl in C++?

Endl is a predefined object of ostream class to insert a new line characters.

53. How to save a file in C++?

When you have written code in the file (notepad),save the file as “hello.cpp.” If you want to write in a file using C++ code, you can do it using iostream and fstream libraries in C++.

```
#include <iostream>
#include <fstream>
using namespace std;
  int main () {
  ofstream file_name;
  file_name.open ("sample.txt");
  file_name<< "Write in the file";
file_name.close();
return 0;
}
```

54. Which operators can be overloaded in C++?

List of operators that can be overloaded are:

+ , – , * , / , % , ^, & , | , ~ , !, =, ++ , –, ==, != , && , ||
+= , -= , /= , %= , ^= , &=, |= , *= , = , [] , (), ->, ->* , new , new [] , delete , delete []

55. How to include all libraries in C++?

The library <bits/stdc++.h> in c++ is used to include all the libraries.

56. How to maximize turbo C++ window?

Alt+Enter is the keyboard shortcut used to maximize (full screen) turbo C++.

57. What is an expression in C++?

An expression is a combination of operators, constants and variables. There seven types of expressions for examples:

– Constant expressions: 89 +10/4.0
– Integral expressions: x * y
– Floating expressions: 17.89
– Relational expressions: a<=b
– Logical expressions: a > b && a == 7
– Pointer expressions: *ptr – Bitwise expressions: p << 5

58. How to write a class in C++?

A class in C++ is the building block that leads to Object-Oriented programming and is a user-defined data type which holds data and functions. The syntax to write a class in C++ is as follows:

```
Class (keyword) Class_Name (this is user defined)
{
   Access specifier: // private, public, protected
Data members //int, char, float, double etc.
   variables to be used
   Member function() { }  // Methods to access
   data members
   };      //Class end
```

For example:

```
    class Sample
    {
        // Access specifier
    private:

        // Data Members
        string s;

        // Member Functions()
    void printname()
        {
            cout << s;
        }
    };
```

59. Which is the best C++ compiler?

There are several good compilers for C++ such as:
– MinGW / GCC
– Borland c++
– Dev C++
– Embracadero
– Clang
– Visual C++
– Intel C++
– Code Block

GCC and clang are great compilers if the programmer's target more portability with good speed.

Intel and other compilers target speed with relatively less emphasis on portability.

60. How to use strcmp function in C++?

strcmp() function is an in-built function of <string.h> header file which takes two strings as arguments and compares these two strings lexicographically.

The syntax of the function is as follows: int strcmp(const char *l, const char *r);

```
#include<stdio.h>
#include<string.h>
int main()
{
    // z has greater ASCII value than
g      char a[] = "zfz";      char b[] =
"gfg";
           int r =
strcmp(a, b);
           if                    (r==0)
printf("Strings are equal");      else
printf("Strings are unequal");

printf("%d" , r);

return 0;
}
```

61. How to write to a file in C++?

A file is read in c++ using a fstream header file.

```
#include <iostream>
#include <fstream>
```

```
    using namespace std;
    int main()
    {
    ofstream fout;
    string r;

    fout.open("test.txt");

    while (fout) {
getline(cin, r);
if (r == "-1")
break;
        fout << line << endl;
    }
    fout.close();

    ifstream fin;
fin.open("test.txt");
while (fin) {
getline(fin, line);
cout << line << endl;
    }
fin.close();
return 0;
}
```

62. What is stringstream in C++?

Stringstream is a class in c++ which associates a string object with a stream allowing to read from the string as if it were a stream. Syntax is as follows:

stringstream string_name(str); Basic

operations are as follows:

clear() str()

<<

>>

63. Why namespace std is used in C++?

If the program does not have using namespace std; then when you write cout <<; you would have to put std::cout <<; same for other functions such as cin, endl etc.

64. How to write hello world in C++?

Hello world in c++ is as follows:

```
#include <iostream>
int main()
{
std::cout << "Hello, World!";
return 0; }
```

65. How to calculate length of a string in C++?

The length of a string can be calculated by using in-built functions such as length(), size(), strlen() and also by loops (while and for).

```
#include<iostream>
#include<cstring>
using namespace std;
main() {
    string s = "Hi I am Mr X";
char arr[] = "Hi I am Mr X";
cout << s.length();    cout <<
s.size();
    cout <<strlen(arr);

    char *c = arr;    int count = 0;
while(*c != '\0'){        count++;
c++;    }    cout << count;    count =
0;    for(int i = 0; arr[i] != '\0';
i++){        count++;
    }
    cout << count;
}
```

66. How to find length of string in C++?

There is an in-built function- length()- in C++ to find the length of the string. The code snippet to find the length of string is as follows

```
string str_1= “abcd”; cout << “The length of the string
is: ” << str_1.length();
```

Advanced C++ Interview Questions

67. What is class in C++?

C language is not an object oriented programming language, so it is a constant attempt of C++ to introduce OOPs. Class is a user defined data type which defines a blueprint of data type. For example,

```
class Circle{

public:
float radius; }
```

68. What is inline function in C++?

Inline functions are functions used to increase the execution time of a program. Basically, if a function is inline, the compiler puts the function code wherever the function is used during compile time. The syntax for the same is as follows:

```
inline return_type function_name(argument list) {
//block of code }
```

69. What is friend function in C++?

A friend function has the access rights to all private and protected members of the class.

```
class Circle{
double radius;

```

```
      public:
      friend void printradius( Circle c );
      };
      void printradius(Circle c ) {
      /* Because printradius() is a friend of
      Circle, it can
      directly access any member of this class
      */
      cout << "Radius of circle: " << c.width;
      }

int main() {
Circle c;
   // Use friend function to print the radius.
   printradius( c);

  return 0;
}
```

70. What is exception handling in C++?

Exceptions are errors that happen during execution of code. To handle them we use throw, try & catch keywords.

71. How to use vector in C++?

A sample code to see how to use vector in C++ is as follows:

```
#include<iostream>
#include<vector>
using namespace std;
int main()
```

```
{
vector <string> vec_1;
vec_1.push_back("sample code");
vec_1.push_back("change example");
for(vector <string>::iterator
i=vec_1.begin();i!=vec_1.end();++i)
cout<<*i;
return 0;
}
```

72. What is vector in C++?

A sequence of containers to store elements, a vector is a template class of C++. Vectors are used when managing ever-changing data elements. The syntax of creating vector.
vector <type> variable (number of elements)

For example:

vector <int> rooms (9);

73. What is scope resolution operator in C++?

Scope resolution operator in c++ is denoted by double colon (::). It can be used:

– when there is a local variable with same name as of global variable
– When a function has to be defined outside a class
– When class’s static variables needs to be accessed
– When a class inside another class has to be referred – In case of multiple Inheritance

74. What are character constants in C++?

Character constant are members of the character set in which a program is written which is surrounded by single quotation marks (‘).

75. What are templates in C++?

A feature that allows functions and classes to operate with generic types that means a function or class can work on different data types without being rewritten is called template.

76. How to sort vector in C++?

```
"#include <bits/stdc++.h>
using namespace std;
int main()
```

```
{
    vector<int> vec{ 1,9,4,3,2,8,5,7};

    sort(vec.begin(), vec.end());

    for (auto x : v)
        cout << x << "" "";

    return 0;
}
"
```

77. What is pure virtual function in C++?

A pure virtual function is a type of virtual function which does not have implementation, but is only declared. It is declared by assigning 0 in declaration. Syntax for the same is as follows:

```
class Test
{
// Data members of class public:

    virtual void show() = 0;

  /* Other members */
};
```

78. How to use map in C++?

Associative containers storing a combination of a key value or mapped value is called Maps. Syntax: map<key_type , value_type> map_name;

```
#include <iostream>
#include <iterator>
```

```
  #include <map>

using namespace std;

  int main()
```

```
{
map<int, int> test;

    // inserting elements
    test.insert(pair<int, int>(1, 2));
    test.insert(pair<int, int>(2, 3));

    map<int, int>::iterator itr;
    for (itr = test.begin(); itr !=
    test.end(); ++itr) {
    cout << itr->first
                  cout <<
    itr->second << '\n';
    }
    return 0;
)
```

79. How to empty a vector in C++?

Std::vector::empty tests whether a vector is empty or not. A sample code for illustrating the same is as follows:

```
#include <iostream>
#include <vector>

  int main ()
  {
  std::vector<int> vec;
  int add (0);

  for (int i=1;i<=5;i++)
  vec.push_back(i);

      while (!vec.empty())
   {
     add+= vec.back();
     vec.pop_back();
```

```
        }

std::cout << add;

  return 0;
```

```
}
```

80. What is visual C++?

C++ is a standardized language and Visual C++ is a product that implements the standard of C++. One can write portable C++ programs using Visual C++, but one can also use Microsoft-only extensions which destroys portability but enhances your productivity.

81. How to remove segmentation fault in C++?

Segmentation fault indicates an error memory corruption. In layman terms, when a piece of code tries to do read and write operation in a read only location in memory. Below are the reasons and solutions for segmentation error:

```
Reason: Accessing an address that is freed int*
p = malloc(8);
*p = 100;
 free(p);
 *p = 110;

Solution: Before freeing the pointer check the assignment or
operation required to perform.

Reason: Accessing out of array index bounds
int arr[2]; arr[3] = 10;
Solution: Correcting the array bound

Reason: Improper use of scanf()
int n = 2;
scanf("%d",n);
Solution: To avoid this is the only solution

Reason: Dereferencing uninitialized pointer int
*p;
printf("%d",*p);
```

```
Solution: A pointer must point to valid memory before
accessi

```

```
Reason: Stack Overflow
   Solution: It can be resolved by having a base condition to
   re the recursive function.
```

82. What is stl in C++ with example?

STL in C++ is a library and abbreviation of Standard Template Library. STL is a generalized library that provides common programming data structures/ container classes, functions, algorithms, and iterators. STL has four components

– Algorithms: Searching and sorting algorithms such as binary search, merge sort etc.
– Containers: Vector, list, queue, arrays, map etc.
– Functions: They are objects that act like functions.
– Iterators: It is an object that allows transversing through elements of a container, e.g., vector<int>::iterator.

83. What is flush in C++? std::flush synchronizes the stream buffer with its controlled output sequence.

84. How to initialize a 2d vector in C++?

The syntax to initialize a 2d vector is as follows: std::vector<std::vector<int> > name_of_vector;

For example: std::vector<std::vector<int> > v { { 1, 2, 1 }, { 2, 6, 7 } };

C++ Programming Interview Questions

85. How to input a string in C++?

There are two ways to input a string in C++. Way 1:

```
string str_1;
cout << "Enter the string";
```

```
cin >> str_1;
```

Way 2:

```
string str_1;
cout << "Enter the string"; getline
(cin, str_1)
```

86. What is virtual function in C++?

A function is said to be virtual if it is defined in base class and is expected to be redefined in derived class.

```
#include <iostream>
using namespace std;

class Base_Class {
public:
    virtual void print_msg() {
cout << "Base";
    }};

class Derived_Class : public Base_Class {
public:
    void print_msg() {
cout << "Derived";
    }
};
```

87. How to find length of array in C++?

The length of an array in C++ can be calculated using sizeof() function. The code depicting the same is mentioned below.

```
#include <iostream>
using namespace std;
```

```

void main()
{
int a[] = {0,1,2,3,4,5};
int a_size = sizeof(a)/sizeof(a[0]);
cout << ""Size of the array is: "" <<
a_size;
}
"
```

88. How to convert int to string in C++?

There are 2 approaches to convert integer variables to string. Both the approaches with a sample code are mentioned below.

Approach 1:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
int n= 1;
string s= to_string(n);
cout << s;
}
```

Approach 2:

```
#include<iostream>
#include  <sstream>
#include   <string>
using namespace std;
int main()
{
    int n = 17;

    // declaring output string stream
ostringstream s1;

    // Sending a number as a stream
into output str     s<< n;
    // the str() converts number into string
string fin = s.str();
    // Displaying the string
    cout << fin;     return 0;
}
```

89. How to sort a string in C++?

To sort a string, the sort function in c++ can be used. The sample code for the same is as follows.

```
    #include<iostream>
    #include <stdio.h>
    using namespace std;
    void str_sort(string &amp;s)
    {
       sort(s.begin(), s.end());
    cout << s;
    }
    int main()
    {
        string s = ""anmbdfc"";
    str_sort(s);       return
    0;
    }

    Output: abcdfmn "
```

90. How to convert string to int in C++?

This can be done using the stoi() or atoi() function:

```
int main()
{
string str_value = "1122";
int int_value =
stoi(str_value);
}
or
```

```
                int main()
                {
                const char *str_ptr = "1122";
                int int_value =
                atoi(str_value);
         }
```

91. How to compare two strings in C++?

Two string can be compared using strcmp() function. It return boolean value; if it returns ‘0’ then the strings are same and if it returns ‘1’ then the two strings are not same.

```
  int main()


{          0) if (strcmp(str_value_a, str_value_b) ==


                same"; cout<<"These strings are the


       else

                the  same";   cout<<"These

  strings are not }
```

92. How to set precision in C++?

Precision in C++ can be set by using the following functions- floor(), ceil(), round(), trunc() and setprecision(). Using setprecision() in a program.

```
#include<bits/stdc++.h> using namespace std;
```

```
void{  main()

      double p = 8.04149;

   pi<<endl; cout << fixed << setprecision(2) <<

   }
```

The output will be 8.04.

93. What is this pointer in C++?

Using this pointer, every object is provided access to its own address. It is an implicit parameter to all member functions. Only member functions have a this pointer and not a friend function.

```
#include<iostream>  using
namespace std;

class sample
{  private:     int a;
public:     void
set_var (int a)
   {
           this->a = a;
   }
   void print()
{
       cout << x ;
     }
};

int main()
{     sample obj;
int a = 2;
obj.set_var(a);
obj.print();
return 0;
}
```

94. How to concatenate two strings in C++?

```
"#include <iostream>
using namespace std;
int main
```

```
{
string str_a = ""Hi, My name is"";
```

```
        string str_b = ""Raj"";
        string str_c = str_a + str_b;
        cout<<""The string is:""<<str_c;
        return 0; 10 }"
```

95. How to find string length in C++?

You can do this by using size() function.

```
            int{   main()

            string str[]=""Hello World!"";
            int g = str.size();

   is""<<g;cout<<""The size of the string

   }
   or you can use the strlen function."
```

96. How to initialize vector in C++?

There are multiple ways to do it:
You can do it like arrays:
vector<int> value{ 11, 22, 33 }; or
by pushing values one by one:
vector<int> value;
value.push_back(11);
value.push_back(22);
value.push_back(33);
value.push_back(44);
value.push_back(55);

97. How to use getline in C++?

```
"int main()
{
string yourname;
getline (cin,yourname);

}"
```

98. How to take input in C++?

You can use the cin function to take in values, like such:

```
int main()
{
      int age;
      cout<<""Enter your name!"";
      cin>>age;
}"
```

99. How to print a string in C++?

```
"int main()
{
string str_value[]=""Hello-World!"";
cout<<""The string is:""<<str_value;
}"
```

100. How to use sort function in C++?

```
"syntax: sort(the element of the array from where
you want to from where you want to finish the
sorting )

int main()
{

int int_value[5]={1,2,3,4,5};

sort(intr_value,intr_value+5);

cout<<intr_value;
}"
```

101. How to reverse a vector in C++?

syntax: reverse(the index you want to start reversing at,the index you want to end reversing at) vector<int> value = {11,22,33}; reverse(value.begin(),value.end());

102. How to return a vector in C++?

```
"#include<iostream>
#include<vector>
```

```

using namespace std;

vector<int> fun(vector<int> value_c)
{
        vector<int> value_d;
        for(int j=0;j<value_c.size();j++)

        value_d.push_back(value_c[j]+4);

        return value_d;
        //returning the vector
}

int main() {                vector<int>
value_a,value_b;

        //We put values in vector value_a
value_a.push_back(1);
value_a.push_back(2);
value_a.push_back(3);

                 value_b =
fun(value_a);
        //receiving the vector

return 0;
}  "
```

103. How to reverse an array in C++?

```
"#include<iostream>
using namespace std;

int main() {
```

```
int n;
cin >> n;
```

```
int a[n];
int i;
for(i = 0; i < n; i++)
{
cin >> arr[i];
}
for(i = n-1; i >= 0; i--)
{
cout << arr[i] << ” “;
}
cout << endl;
return 0;

}
"
```

104. How to split a string in C++?

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

   int main ()
   {
   char * flag_pointer;
   char string_value[10]=""Great Learning!"";
   flag_pointer =strtok (string_value,"" !"");

   while (flag_pointer!=NULL)
   {
      cout<<string_value;
      flag_pointer=strtok(NULL,"" !"");
      }

 return 0;
}"
```

105. How to sort string in C++?

To sort a string, the sort function in c++ can be used. The sample code for the same is as follows.

```
#include<iostream>
#include <stdio.h>
using namespace std;
void str_sort(string &amp;s)  {
    sort(s.begin(), s.end());
    cout << s;
    }

     int main()
     {
     string s = ""anmbdfc"";
     str_sort(s);
     return 0;
     }

Output: abcdfmn "
```

OOPS Interview Questions C++

106. What is constructor in C++?

Constructor is a method in class which has the same name as that of class and is followed by parentheses (). It is automatically called when an object of a class is created.

```
class Hello {      // The class

        public:              // Access specifier

        Hello() {      // Constructor
cout << ""Hello World!"";

}

};

main() {
int
Hello obj;     // Create an object of Hello
(this will

call the constructor) 0;

return }

"
```

107. What is inheritance in C++?

Just like a child inherits some features and attributes from his parent similarly a class inherit attributes and methods from another class. The parent class is called base class and the child class is called derived class.

```
// Base class
class Food_Item{
public:
    void taste() {      cout << ""The taste of
every food item is different. \n"";
  }};

// Derived class class Chips:
public Food_Item{  public:
    void taste() {     cout << ""The taste of
chips is salty \n""
;   }
};
"
```

108. What is object in C++?

Class in C++ provides a blueprint for object, that means, object is created from the class.

```
For example,
class Circle{
         public:
         float radius;
         }

Circle C1;
Circle C2;
```

30 **109. What is encapsulation in C++?**

To prevent access to data directly, Encapsulation is the process that combines data variables and functions in a class. This is achieved by doing the following:

1. Making all data variables private.
2. Creating getter and setter functions for data variables.

110. What is abstraction in C++?

Abstraction in C++ means showing only what is necessary. It’s part of Object oriented Programming concept. Abstraction is used to hide any irrelevant data to the outside world and only showing what is absolutely necessary for the outside world to use. eg. Classes use the abstraction concept to only show relevant data types or elements. This is done through access specifiers such as: public, private, protected.

111. What is oops in C++?

OOP or Object Oriented Programming in C++ is a type of programming in which you create objects and classes to emulate real world concepts such as Abstraction, Polymorphism, Encapsulation, Inheritance.
Here classes are data types that allow you list several types of data within it and even functions. You can access these classes with the help of class objects.

112. What is member function in C++?

Member functions are those functions that you declare within a class, they are members of the class. You can reference them using class objects. Eg.

```
class A
{
        public:
        int add(int b)
        {
        a = b * 10;
        return a;
        };
};"
```

113. What is virtual base class in C++?

Let’s understand this with an example.

You Have 4 classes: W,X,Y,Z
Here X & Y inherit from W. So they both have similar features being inherited from W.
Now, Z inherits from both X & Y

Here Z may inherit similar features from X & Y as they both have inherited them from W. This can cause issues and that’s why we use virtual base classes as they stop multiple features of a class from appearing in another class.

114. How to access private members of a class in C++?

Private members of the class are not accessible by object or function outside the class. Only functions inside the class can access them or friend functions. However, pointers can be used to access private data members outside the class. Sample code is as follows:

```
#include <iostream>  using
namespace std;

class sample_test{
private:      int
n;

public:      sample_test() { n
= 45; }       int display() {
return n;
      }
};
```

115. How to call base class constructor from derived class in C++?

A base class constructor will be called whenever the derived class constructor is called. Upon the creation of a dervied class object the order of constructor execution is : base class constructor then Default class constructor.

116. What is an abstract class in C++?

An abstract class in C++ is such that cannot be used directly and is used to form a base class for others to inherit from.
If you create an object for an abstract class the compiler will throw an error at you.

117. What is containership in C++?

Containership in C++ is a relationship in which a class’s object is nested within another class. The class that contains the object is called a container class and the class whose object is stored is called a contained class.

118. What is data hiding in C++?

An object oriented technique of hiding data members is called data hiding. In other words, giving restricted access to the data members so as to maintain object integrity.

119. What is runtime polymorphism in C++?

Polymorphism means having many forms either it is a function or operator in programming.
Runtime polymorphism is achieved by function overriding.

```
#include <bits/stdc++.h>  using
namespace std;
class parent
{  public:
void print()
    { cout<< ""base class""; }
};
class child:public parent
{
public:
void
print()
    { cout<< ""derived class""; }
};
```

```
int main()
{      parent
*p;      child
c;      p =
&c;  //virtual
function,
binded at
runtime
(Runtime  polymorphism)
p->print();      return
0;
}
```

120. What is copy constructor in C++?

A copy constructor is a constructor which creates an object by initializing it with an object of the same class, which has been created previously. The syntax for copy constructor is as follows:

classname (const classname &obj) {
// body of constructor }

121. How is modularity introduced in C++?

Modularity is a way of mapping encapsulated abstractions into real and physical modules which is closely related to Encapsulation. It is a concept in which separate programs are divided into separate modules.

For example, when building a house it is built in modular way. First foundation is laid, then structure is made and so on.

122. What is the size of empty class in C++? Size of an empty class is 1 byte generally just to ensure that the two different objects will have different addresses.