C++ Interview Questions

Dear readers, these **C++ Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **C++**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer −

What is the full form of OOPS?

Object Oriented Programming System.

What is a class?

Class is a blue print which reflects the entities attributes and actions. Technically defining a class is designing an user defined data type.

What is an object?

An instance of the class is called as object.

List the types of inheritance supported in C++.

Single, Multilevel, Multiple, Hierarchical and Hybrid.

What is the role of protected access specifier?

If a class member is protected then it is accessible in the inherited class. However, outside the both the private and protected members are not accessible.

What is encapsulation?

The process of binding the data and the functions acting on the data together in an entity (class) called as encapsulation.

What is abstraction?

Abstraction refers to hiding the internal implementation and exhibiting only the necessary details.

What is inheritance?

Inheritance is the process of acquiring the properties of the exiting class into the new class. The existing class is called as base/parent class and the inherited class is called as derived/child class.

Explain the purpose of the keyword volatile.

Declaring a variable volatile directs the compiler that the variable can be changed externally. Hence avoiding compiler optimization on the variable reference.

What is an inline function?

A function prefixed with the keyword inline before the function definition is called as inline function. The inline functions are faster in execution when compared to normal functions as the compiler treats inline functions as macros.

What is a storage class?

Storage class specifies the life or scope of symbols such as variable or functions.

Mention the storage classes names in C++.

The following are storage classes supported in C++

auto, static, extern, register and mutable

What is the role of mutable storage class specifier?

A constant class object’s member variable can be altered by declaring it using mutable storage class specifier. Applicable only for non-static and non-constant member variable of the class.

Distinguish between shallow copy and deep copy.

Shallow copy does memory dumping bit-by-bit from one object to another. Deep copy is copy field by field from object to another. Deep copy is achieved using copy constructor and or overloading assignment operator.

What is a pure virtual function?

A virtual function with no function body and assigned with a value zero is called as pure virtual function.

What is an abstract class in C++?

A class with at least one pure virtual function is called as abstract class. We cannot instantiate an abstract class.

What is a reference variable in C++?

A reference variable is an alias name for the existing variable. Which mean both the variable name and reference variable point to the same memory location. Therefore updation on the original variable can be achieved using reference variable too.

What is role of static keyword on class member variable?

A static variable does exit though the objects for the respective class are not created. Static member variable share a common memory across all the objects created for the respective class. A static member variable can be referred using the class name itself.

Explain the static member function.

A static member function can be invoked using the class name as it exits before class objects comes into existence. It can access only static members of the class.

Name the data type which can be used to store wide characters in C++.

wchar_t

What are/is the operator/operators used to access the class members?

Dot (.) and Arrow (->)

Can we initialize a class/structure member variable as soon as the same is defined?

No, Defining a class/structure is just a type definition and will not allocated memory for the same.

What is the data type to store the Boolean value?

bool, is the new primitive data type introduced in C++ language.

What is function overloading?

Defining several functions with the same name with unique list of parameters is called as function overloading.

What is operator overloading?

Defining a new job for the existing operator w.r.t the class objects is called as operator overloading.

Do we have a String primitive data type in C++?

No, it’s a class from STL (Standard template library).

Name the default standard streams in C++.

cin, cout, cerr and clog.

Which access specifier/s can help to achive data hiding in C++?

Private & Protected.

When a class member is defined outside the class, which operator can be used to associate the function definition to a particular class?

Scope resolution operator (::)

What is a destructor? Can it be overloaded?

A destructor is the member function of the class which is having the same name as the class name and prefixed with tilde (~) symbol. It gets executed automatically w.r.t the object as soon as the object loses its scope. It cannot be overloaded and the only form is without the parameters.

What is a constructor?

A constructor is the member function of the class which is having the same as the class name and gets executed automatically as soon as the object for the respective class is created.

What is a default constructor? Can we provide one for our class?

Every class does have a constructor provided by the compiler if the programmer doesn’t provides one and known as default constructor. A programmer provided constructor with no parameters is called as default constructor. In such case compiler doesn’t provides the constructor.

Which operator can be used in C++ to allocate dynamic memory?

‘new’ is the operator can be used for the same.

What is the purpose of ‘delete’ operator?

‘delete’ operator is used to release the dynamic memory which was created using ‘new’ operator.

Can I use malloc() function of C language to allocate dynamic memory in C++?

Yes, as C is the subset of C++, we can all the functions of C in C++ too.

Can I use ‘delete’ operator to release the memory which was allocated using malloc() function of C language?

No, we need to use free() of C language for the same.

What is a friend function?

A function which is not a member of the class but still can access all the member of the class is called so. To make it happen we need to declare within the required class following the keyword ‘friend’.

What is a copy constructor?

A copy constructor is the constructor which take same class object reference as the parameter. It gets automatically invoked as soon as the object is initialized with another object of the same class at the time of its creation.

Does C++ supports exception handling? If so what are the keywords involved in achieving the same.

C++ does supports exception handling. try, catch & throw are keyword used for the same.

Explain the pointer – this.

This, is the pointer variable of the compiler which always holds the current active object’s address.

What is the difference between the keywords struct and class in C++?

By default the members of struct are public and by default the members of the class are private.

Can we implement all the concepts of OOPS using the keyword struct?

Yes.

What is the block scope variable in C++?

A variable whose scope is applicable only within a block is said so. Also a variable in C++ can be declared anywhere within the block.

What is the role of the file opening mode ios::trunk?

If the file already exists, its content will be truncated before opening the file.

What is the scope resolution operator?

The scope resolution operator is used to

- Resolve the scope of global variables.
- To associate function definition to a class if the function is defined outside the class.

What is a namespace?

A namespace is the logical division of the code which can be used to resolve the name conflict of the identifiers by placing them under different name space.

What are command line arguments?

The arguments/parameters which are sent to the main() function while executing from the command line/console are called so. All the arguments sent are the strings only.

What is a class template?

A template class is a generic class. The keyword template can be used to define a class template.

How can we catch all kind of exceptions in a single catch block?

The catch block with ellipses as follows

```
catch(…)
{
}
```

What is keyword auto for?

By default every local variable of the function is automatic (auto). In the below function both the variables ‘i’ and ‘j’ are automatic variables.

```
void f()
{
   int i;

   auto int j;
}
```

NOTE − A global variable can't be an automatic variable.

What is a static variable?

A static local variables retains its value between the function call and the default value is 0. The following function will print 1 2 3 if called thrice.

```
void f()
{
   static int i;

   ++i;
   printf("%d ",i);
}
```

If a global variable is static then its visibility is limited to the same source code.

What is the purpose of extern storage specifier.

Used to resolve the scope of global symbol

```
#include <iostream>

using namespace std;
main() {
   extern int i;

   cout<<i<<endl;
}
int i = 20;
```

What is the meaning of base address of the array?

The starting address of the array is called as the base address of the array.

When should we use the register storage specifier?

If a variable is used most frequently then it should be declared using register storage specifier, then possibly the compiler gives CPU register for its storage to speed up the look up of the variable.

Can a program be compiled without main() function?

Yes, it can be but cannot be executed, as the execution requires main() function definition.

Where an automatic variable is stored?

Every local variable by default being an auto variable is stored in stack memory

What is a container class?

A class containing at least one member variable of another class type in it is called so.

What is a token?

A C++ program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.

What is a preprocessor?

Preprocessor is a directive to the compiler to perform certain things before the actual compilation process begins.

What are command line arguments?

The arguments which we pass to the main() function while executing the program are called as command line arguments. The parameters are always strings held in the second argument (below in args) of the function which is array of character pointers. First argument represents the count of arguments (below in count) and updated automatically by operating system.

```
main( int count, char *args[]) {
}
```

What are the different ways of passing parameters to the functions? Which to use when?

- **Call by value** − We send only values to the function as parameters. We choose this if we do not want the actual parameters to be modified with formal parameters but just used.
- **Call by address** − We send address of the actual parameters instead of values. We choose this if we do want the actual parameters to be modified with formal parameters.
- **Call by reference** − The actual parameters are received with the C++ new reference variables as formal parameters. We choose this if we do want the actual parameters to be modified with formal parameters.

What is reminder for 5.0 % 2?

Error, It is invalid that either of the operands for the modulus operator (%) is a real number.

Which compiler switch to be used for compiling the programs using math library with g++ compiler?

Opiton –lm to be used as > g++ –lm <file.cpp>

Can we resize the allocated memory which was allocated using ‘new’ operator?

No, there is no such provision available.

Who designed C++ programming language?

Bjarne Stroustrup.

Which operator can be used to determine the size of a data type/class or variable/object?

sizeof

How can we refer to the global variable if the local and the global variable names are same?

We can apply scope resolution operator (::) to the for the scope of global variable.

What are valid operations on pointers?

The only two permitted operations on pointers are

- Comparision ii) Addition/Substraction (excluding void pointers)

What is recursion?

Function calling itself is called as recursion.

What is the first string in the argument vector w.r.t command line arguments?

Program name.

What is the maximum length of an identifier?

Ideally it is 32 characters and also implementation dependent.

What is the default function call method?

By default the functions are called by value.

What are available mode of inheritance to inherit one class from another?

Public, private & protected

What is the difference between delete and delete[]?

Delete[] is used to release the array allocated memory which was allocated using new[] and delete is used to release one chunk of memory which was allocated using new.

Does an abstract class in C++ need to hold all pure virtual functions?

Not necessarily, a class having at least one pure virtual function is abstract class too.

Is it legal to assign a base class object to a derived class pointer?

No, it will be error as the compiler fails to do conversion.

What happens if an exception is thrown outside a try block?

The program shall quit abruptly.

Are the exceptions and error same?

No, exceptions can be handled whereas program cannot resolve errors.

What is function overriding?

Defining the functions within the base and derived class with the same signature and name where the base class's function is virtual.

Which function is used to move the stream pointer for the purpose of reading data from stream?

seekg()

Which function is used to move the stream pointer for the purpose of writing data from stream?

seekp()

Are class functions taken into consideration as part of the object size?

No, only the class member variables determines the size of the respective class object.

Can we create and empty class? If so what would be the size of such object.

We can create an empty class and the object size will be 1.

What is ‘std’?

Default namespace defined by C++.

What is the full form of STL?

Standard template library

What is ‘cout’?

cout is the object of ostream class. The stream ‘cout’ is by default connected to console output device.

What is ‘cin’?

cin is the object of istream class. The stream ‘cin’ is by default connected to console input device.

What is the use of the keyword ‘using’?

It is used to specify the namespace being used in.

If a pointer declared for a class, which operator can be used to access its class members?

Arrow (->) operator can be used for the same

What is difference between including the header file with-in angular braces < > and double quotes “ “

If a header file is included with in < > then the compiler searches for the particular header file only with in the built in include path. If a header file is included with in “ “, then the compiler searches for the particular header file first in the current working directory, if not found then in the built in include path

S++ or S = S+1, which can be recommended to increment the value by 1 and why?

S++, as it is single machine instruction (INC) internally.

What is the difference between actual and formal parameters?

The parameters sent to the function at calling end are called as actual parameters while at the receiving of the function definition called as formal parameters.

What is the difference between variable declaration and variable definition?

Declaration associates type to the variable whereas definition gives the value to the variable.

Which key word is used to perform unconditional branching?

goto.

Is 068 a valid octal number?

No, it contains invalid octal digits.

What is the purpose of #undef preprocessor?

It will be used to undefine an existing macro definition.

Can we nest multi line comments in a C++ code?

No, we cannot.

What is a virtual destructor?

A virtual destructor ensures that the objects resources are released in the reverse order of the object being constructed w.r.t inherited object.

What is the order of objects destroyed in the memory?

The objects are destroyed in the reverse order of their creation.

What is a friend class?

A class members can gain accessibility over other class member by placing the class declaration prefixed with the keyword ‘friend’ in the destination class.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not

expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

C++ Interview Questions

A list of top frequently asked C++ interview questions and answers are given below.

1) What is C++?

C++ is an object-oriented programming language created by Bjarne Stroustrup. It was released in 1985.

C++ is a superset of C with the major addition of classes in C language.

Initially, Stroustrup called the new language "C with classes". However, after sometime the name was changed to C++. The idea of C++ comes from the C increment operator ++.

2) What are the advantages of C++?

C++ doesn't only maintains all aspects from C language, it also simplifies memory management and adds several features like:

- C++ is a highly portable language means that the software developed using C++ language can run on any platform.
- C++ is an object-oriented programming language which includes the concepts such as classes, objects, inheritance, polymorphism, abstraction.
- C++ has the concept of inheritance. Through inheritance, one can eliminate the redundant code and can reuse the existing classes.
- Data hiding helps the programmer to build secure programs so that the program cannot be attacked by the invaders.
- Message passing is a technique used for communication between the objects.
- C++ contains a rich function library.

3) What is the difference between C and C++?

Following are the differences between C and C++:

C	C++
C language was developed by Dennis Ritchie.	C++ language was developed by Bjarne Stroustrup.
C is a structured programming language.	C++ supports both structural and object-oriented programming language.
C is a subset of C++.	C++ is a superset of C.
In C language, data and functions are the free entities.	In the C++ language, both data and functions are encapsulated together in the form of a project.
C does not support the data hiding. Therefore, the data can be used by the outside world.	C++ supports data hiding. Therefore, the data cannot be accessed by the outside world.
C supports neither function nor operator overloading.	C++ supports both function and operator overloading.
In C, the function cannot be implemented inside the structures.	In the C++, the function can be implemented inside the structures.
Reference variables are not supported in C language.	C++ supports the reference variables.
C language does not support the virtual and friend functions.	C++ supports both virtual and friend functions.
In C, scanf() and printf() are mainly used for input/output.	C++ mainly uses stream cin and cout to perform input and output operations.

4) What is the difference between reference and pointer?

Following are the differences between reference and pointer:

Reference	Pointer
Reference behaves like an alias for an existing variable, i.e., it is a temporary variable.	The pointer is a variable which stores the address of a variable.
Reference variable does not require any indirection operator to access the value. A reference variable can be used directly to access the value.	Pointer variable requires an indirection operator to access the value of a variable.
Once the reference variable is assigned, then it cannot be reassigned with different address values.	The pointer variable is an independent variable means that it can be reassigned to point to different objects.
A null value cannot be assigned to the reference variable.	A null value can be assigned to the reference variable.
It is necessary to initialize the variable at the time of declaration.	It is not necessary to initialize the variable at the time of declaration.

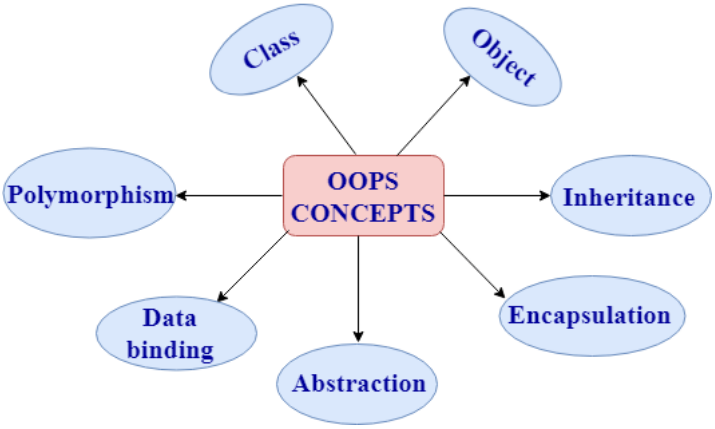
5) What is a class?

The class is a user-defined data type. The class is declared with the keyword class. The class contains the data members, and member functions whose access is defined by the three modifiers are private, public and protected. The class defines the type definition of the category of things. It defines a datatype, but it does not define the data it just specifies the structure of data.

You can create N number of objects from a class.

6) What are the various OOPs concepts in C++?

The various OOPS concepts in C++ are:



- **Class:**

The class is a user-defined data type which defines its properties and its functions. For example, Human being is a class. The body parts of a human being are its properties, and the actions performed by the body parts are known as functions. The class does not occupy any memory space. Therefore, we can say that the class is the only logical representation of the data.

The syntax of declaring the class:

```
class student
{
//data members;
//Member functions
}
```

- **Object:**

An object is a run-time entity. An object is the instance of the class. An object can represent a person, place or any other item. An object can operate on both data members and member functions. The class does not occupy any memory space. When an object is created using a new keyword, then space is allocated for the variable in a heap, and the starting address is stored in the stack memory. When an object is created without a new keyword, then space is not allocated in the heap memory, and the object contains the null value in the stack.

```
class Student
{
//data members;
//Member functions
}
```

The syntax for declaring the object:

```
Student s = new Student();
```

- **Inheritance:**

Inheritance provides reusability. Reusability means that one can use the functionalities of the existing class. It eliminates the redundancy of code. Inheritance is a technique of deriving a new class from the old class. The old class is known as the base class, and the new class is known as derived class.

Syntax

```
class derived_class :: visibility-mode base_class;
```

> Note: The visibility-mode can be public, private, protected.

- **Encapsulation:**

Encapsulation is a technique of wrapping the data members and member functions in a single unit. It binds the data within a class, and no outside method can access the data. If the data member is private, then the member function can only access the data.

- **Abstraction:**

Abstraction is a technique of showing only essential details without representing the implementation details. If the members are defined with a public keyword, then the members are accessible outside also. If the members are defined with a private keyword, then the members are not accessible by the outside methods.

- **Data binding:**

Data binding is a process of binding the application UI and business logic. Any change made in the business logic will reflect directly to the application UI.

- **Polymorphism:**

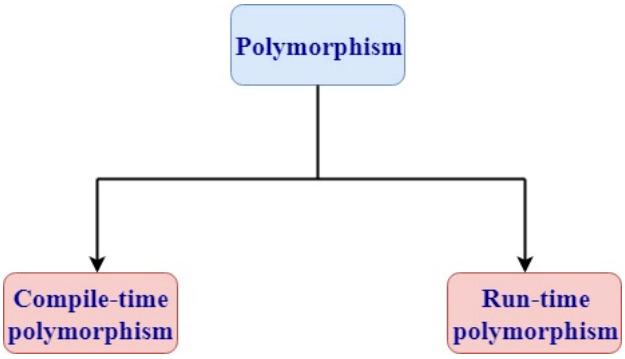
Polymorphism means multiple forms. Polymorphism means having more than one function with the same name but with different functionalities. Polymorphism is of two types:

1. Static polymorphism is also known as early binding.
2. Dynamic polymorphism is also known as late binding.

7) What are the different types of polymorphism in C++?

Polymorphism: Polymorphism means multiple forms. It means having more than one function with the same function name but with different functionalities.

Polymorphism is of two types:



- **Runtime polymorphism**

Runtime polymorphism is also known as dynamic polymorphism. Function overriding is an example of runtime polymorphism. Function overriding means when the child class contains the method which is already present in the parent class. Hence, the child class overrides the method

⇧

of the parent class. In case of function overriding, parent and child class both contains the same function with the different definition. The call to the function is determined at runtime is known as runtime polymorphism.

Let's understand this through an example:

```
#include <iostream>
using namespace std;
class Base
{
   public:
   virtual void show()
   {
      cout<<"javaTpoint";
    }
};
class Derived:public Base
{
   public:
   void show()
   {
      cout<<"javaTpoint tutorial";
   }
};

int main()
{
   Base* b;
   Derived d;
   b=&d;
   b->show();
            return 0;
}
```

Output:

```
javaTpoint tutorial
```

- **Compile time polymorphism**

Compile-time polymorphism is also known as static polymorphism. The polymorphism which is implemented at the compile time is known as compile-time polymorphism. Method overloading is an example of compile-time polymorphism.

Method overloading: Method overloading is a technique which allows you to have more than one function with the same function name but with different functionality.

Method overloading can be possible on the following basis:

- The return type of the overloaded function.
- The type of the parameters passed to the function.
- The number of parameters passed to the function.

Let's understand this through an example:

```
#include <iostream>
using namespace std;
class Multiply
{
```

```
  public:
  int mul(int a,int b)
  {
     return(a*b);
  }
  int mul(int a,int b,int c)
  {
     return(a*b*c);
  }
 };
int main()
{
   Multiply multi;
   int res1,res2;
   res1=multi.mul(2,3);
   res2=multi.mul(2,3,4);
   cout<<"\n";
   cout<<res1;
   cout<<"\n";
   cout<<res2;
   return 0;
}
```

Output:

```
6
24
```

- In the above example, mul() is an overloaded function with the different number of parameters.

8) Define namespace in C++.

- The namespace is a logical division of the code which is designed to stop the naming conflict.
- The namespace defines the scope where the identifiers such as variables, class, functions are declared.
- The main purpose of using namespace in C++ is to remove the ambiguity. Ambiquity occurs when the different task occurs with the same name.
- For example: if there are two functions exist with the same name such as add(). In order to prevent this ambiguity, the namespace is used. Functions are declared in different namespaces.
- C++ consists of a standard namespace, i.e., std which contains inbuilt classes and functions. So, by using the statement "using namespace std;" includes the namespace "std" in our program.
- **Syntax of namespace:**

```
namespace namespace_name
{
 //body of namespace;
}
```

Syntax of accessing the namespace variable:

```
namespace_name::member_name;
```

Let's understand this through an example:

```
#include <iostream>
using namespace std;
namespace addition
{
   int a=5;
   int b=5;
   int add()
   {
      return(a+b);
   }
}

int main() {
int result;
result=addition::add();
cout<<result;
return 0;
}
```

Output:

```
10
```

9) Define token in C++.

A token in C++ can be a keyword, identifier, literal, constant and symbol.

10) Who was the creator of C++?

Bjarne Stroustrup.

11) Which operations are permitted on pointers?

Following are the operations that can be performed on pointers:

- **Incrementing or decrementing a pointer**: Incrementing a pointer means that we can increment the pointer by the size of a data type to which it points.

There are two types of increment pointers:

1. Pre-increment pointer: The pre-increment operator increments the operand by 1, and the value of the expression becomes the resulting value of the incremented. Suppose ptr is a pointer then pre-increment pointer is represented as ++ptr.

Let's understand this through an example:

```
#include <iostream>
using namespace std;
int main()
{
int a[5]={1,2,3,4,5};
int *ptr;
ptr=&a[0];
cout<<"Value of *ptr is : "<<*ptr<<"\n";
cout<<"Value of *++ptr : "<<*++ptr;
return 0;
```

```
}
```

Output:

```
Value of *ptr is : 1
Value of *++ptr : 2
```

2. Post-increment pointer: The post-increment operator increments the operand by 1, but the value of the expression will be the value of the operand prior to the incremented value of the operand. Suppose ptr is a pointer then post-increment pointer is represented as ptr++.

Let's understand this through an example:

```
#include <iostream>
using namespace std;
int main()
{
int a[5]={1,2,3,4,5};
int *ptr;
ptr=&a[0];
cout<<"Value of *ptr is : "<<*ptr<<"\n";
cout<<"Value of *ptr++ : "<<*ptr++;
return 0;
}
```

Output:

```
Value of *ptr is : 1
Value of *ptr++ : 1
```

- **Subtracting a pointer from another pointer:** When two pointers pointing to the members of an array are subtracted, then the number of elements present between the two members are returned.

12) Define 'std'.

Std is the default namespace standard used in C++.

13) Which programming language's unsatisfactory performance led to the discovery of C++?

C++was discovered in order to cope with the disadvantages of C.

14) How delete [] is different from delete?

Delete is used to release a unit of memory, delete[] is used to release an array.

15) What is the full form of STL in C++?

STL stands for Standard Template Library.

16) What is an object?

The Object is the instance of a class. A class provides a blueprint for objects. So you can create an object from a class. The objects of a class are declared with the same sort of declaration that we declare variables of basic types.

17) What are the C++ access specifiers?

The access specifiers are used to define how to functions and variables can be accessed outside the class.

There are three types of access specifiers:

- **Private**: Functions and variables declared as private can be accessed only within the same class, and they cannot be accessed outside the class they are declared.
- **Public**: Functions and variables declared under public can be accessed from anywhere.
- **Protected**: Functions and variables declared as protected cannot be accessed outside the class except a child class. This specifier is generally used in inheritance.

18) What is Object Oriented Programming (OOP)?

OOP is a methodology or paradigm that provides many concepts. The basic concepts of Object Oriented Programming are given below:

Classes and Objects: Classes are used to specify the structure of the data. They define the data type. You can create any number of objects from a class. Objects are the instances of classes.

Encapsulation: Encapsulation is a mechanism which binds the data and associated operations together and thus hides the data from the outside world. Encapsulation is also known as data hiding. In C++, It is achieved using the access specifiers, i.e., public, private and protected.

Abstraction: Abstraction is used to hide the internal implementations and show only the necessary details to the outer world. Data abstraction is implemented using interfaces and abstract classes in C++.

Some people confused about Encapsulation and abstraction, but they both are different.

Inheritance: Inheritance is used to inherit the property of one class into another class. It facilitates you to define one class in term of another class.

19) What is the difference between an array and a list?

- An Array is a collection of homogeneous elements while a list is a collection of heterogeneous elements.
- Array memory allocation is static and continuous while List memory allocation is dynamic and random.
- In Array, users don't need to keep in track of next memory allocation while In the list, the user has to keep in track of next location where memory is allocated.

20) What is the difference between new() and malloc()?

- new() is a preprocessor while malloc() is a function.
- There is no need to allocate the memory while using "new" but in malloc() you have to use sizeof().
- "new" initializes the new memory to 0 while malloc() gives random value in the newly allotted memory location.
- The new() operator allocates the memory and calls the constructor for the object initialization and malloc() function allocates the memory but does not call the constructor for the object initialization.
- The new() operator is faster than the malloc() function as operator is faster than the function.

21) What are the methods of exporting a function from a DLL?

There are two ways:

- By using the DLL's type library.

⇧

- Taking a reference to the function from the DLL instance.

22) Define friend function.

Friend function acts as a friend of the class. It can access the private and protected members of the class. The friend function is not a member of the class, but it must be listed in the class definition. The non-member function cannot access the private data of the class. Sometimes, it is necessary for the non-member function to access the data. The friend function is a non-member function and has the ability to access the private data of the class.

To make an outside function friendly to the class, we need to declare the function as a friend of the class as shown below:

```
class sample
{
  // data members;
 public:
friend void abc(void);
};
```

Following are the characteristics of a friend function:

- The friend function is not in the scope of the class in which it has been declared.
- Since it is not in the scope of the class, so it cannot be called by using the object of the class. Therefore, friend function can be invoked like a normal function.
- A friend function cannot access the private members directly, it has to use an object name and dot operator with each member name.
- Friend function uses objects as arguments.

Let's understand this through an example:

```
#include <iostream>
using namespace std;
class Addition
{
 int a=5;
 int b=6;
 public:
 friend int add(Addition a1)
 {
    return(a1.a+a1.b);
 }
};
int main()
{
int result;
Addition a1;
 result=add(a1);
 cout<<result;
return 0;
}
```

Output:

```
11
```

23) What is a virtual function?

- A virtual function is used to replace the implementation provided by the base class. The replacement is always called whenever the object in question is actually of the derived class, even if the object is accessed by a base pointer rather than a derived pointer.
- A virtual function is a member function which is present in the base class and redefined by the derived class.
- When we use the same function name in both base and derived class, the function in base class is declared with a keyword virtual.
- When the function is made virtual, then C++ determines at run-time which function is to be called based on the type of the object pointed by the base class pointer. Thus, by making the base class pointer to point different objects, we can execute different versions of the virtual functions.

Rules of a virtual function:

- The virtual functions should be a member of some class.
- The virtual function cannot be a static member.
- Virtual functions are called by using the object pointer.
- It can be a friend of another class.
- C++ does not contain virtual constructors but can have a virtual destructor.

24) When should we use multiple inheritance?

You can answer this question in three manners:

1. Never
2. Rarely
3. If you find that the problem domain cannot be accurately modeled any other way.

25) What is a destructor?

A Destructor is used to delete any extra resources allocated by the object. A destructor function is called automatically once the object goes out of the scope.

Rules of destructor:

- Destructors have the same name as class name and it is preceded by tilde.
- It does not contain any argument and no return type.

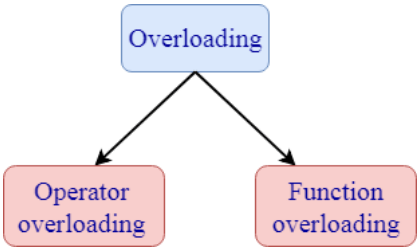
26) What is an overflow error?

It is a type of arithmetical error. It happens when the result of an arithmetical operation been greater than the actual space provided by the system.

27) What is overloading?

- When a single object behaves in many ways is known as overloading. A single object has the same name, but it provides different versions of the same function.
- C++ facilitates you to specify more than one definition for a function name or an operator in the same scope. It is called function overloading and operator overloading respectively.
- **Overloading is of two types:**

⇧



1. Operator overloading: Operator overloading is a compile-time polymorphism in which a standard operator is overloaded to provide a user-defined definition to it. For example, '+' operator is overloaded to perform the addition operation on data types such as int, float, etc.

Operator overloading can be implemented in the following functions:

- Member function
- Non-member function
- Friend function

Syntax of Operator overloading:

```
Return_type classname :: Operator Operator_symbol(argument_list)
{
    // body_statements;
}
```

2. Function overloading: Function overloading is also a type of compile-time polymorphism which can define a family of functions with the same name. The function would perform different operations based on the argument list in the function call. The function to be invoked depends on the number of arguments and the type of the arguments in the argument list.

28) What is function overriding?

If you inherit a class into a derived class and provide a definition for one of the base class's function again inside the derived class, then this function is called overridden function, and this mechanism is known as function overriding.

29) What is virtual inheritance?

Virtual inheritance facilitates you to create only one copy of each object even if the object appears more than one in the hierarchy.

30) What is a constructor?

A Constructor is a special method that initializes an object. Its name must be same as class name.

31) What is the purpose of the "delete" operator?

The "delete" operator is used to release the dynamic memory created by "new" operator.

32) Explain this pointer?

This pointer holds the address of the current object.

33) What does Scope Resolution operator do?

A scope resolution operator(::) is used to define the member function outside the class.

34) What is the difference between delete and delete[]?

Delete [] is used to release the array of allocated memory which was allocated using new[] whereas delete is used to release one chunk of memory which was allocated using new.

35) What is a pure virtual function?

The pure virtual function is a virtual function which does not contain any definition. The normal function is preceded with a keyword virtual. The pure virtual function ends with 0.

Syntax of a pure virtual function:

```
virtual void abc()=0;   //pure virtual function.
```

Let's understand this through an example:

```
#include<iostream>
using namespace std;
class Base
{
   public:
   virtual void show()=0;
};

class Derived:public Base
{
   public:
   void show()
   {
      cout<<"javaTpoint";
   }
};
int main()
{
   Base* b;
   Derived d;
   b=&d;
   b->show();
   return 0;
}
```

Output:

```
javaTpoint
```

36) What is the difference between struct and class?

Structures	class
A structure is a user-defined data type which contains variables of dissimilar data types.	The class is a user-defined data type which contains member variables and member functions.
The variables of a structure are stored in the stack memory.	The variables of a class are stored in the heap memory.
We cannot initialize the variables directly.	We can initialize the member variables directly.

If access specifier is not specified, then by default the access specifier of the variable is "public".	If access specifier is not specified, then by default the access specifier of a variable is "private".
The instance of a structure is a "structure variable".	

Declaration of a structure:
`struct structure_name`
`{`
`  // body of structure;`
`} ;`
	Declaration of class:
	`class class_name`
	`{`
	`  // body of class;`
	`}`

A structure is declared by using a struct keyword.	The class is declared by using a class keyword.
The structure does not support the inheritance.	The class supports the concept of inheritance.
The type of a structure is a value type.	The type of a class is a reference type.

37) What is a class template?

A class template is used to create a family of classes and functions. For example, we can create a template of an array class which will enable us to create an array of various types such as int, float, char, etc. Similarly, we can create a template for a function, suppose we have a function add(), then we can create multiple versions of add().

The syntax of a class template:

```
template<class T>
class classname
{
  // body of class;
};
```

Syntax of a object of a template class:

```
classname<type> objectname(arglist);
```

38) What is the difference between function overloading and operator overloading?

Function overloading: Function overloading is defined as we can have more than one version of the same function. The versions of a function will have different signature means that they have a different set of parameters.

Operator overloading: Operator overloading is defined as the standard operator can be redefined so that it has a different meaning when applied to the instances of a class.

39) What is a virtual destructor?

A virtual destructor in C++ is used in the base class so that the derived class object can also be destroyed. A virtual destructor is declared by using the ~ tilde operator and then virtual keyword before the constructor.

> Note: Constructor cannot be virtual, but destructor can be virtual.

Let's understand this through an example

- Example without using virtual destructor

```
#include <iostream>
using namespace std;
class Base
{
   public:
   Base()
   {
      cout<<"Base constructor is called"<<"\n";
   }
   ~Base()
   {
      cout<<"Base class object is destroyed"<<"\n";
   }
};
class Derived:public Base
{
   public:
   Derived()
   {
      cout<<"Derived class constructor is called"<<"\n";
   }
   ~Derived()
   {
      cout<<"Derived class object is destroyed"<<"\n";
   }
};
int main()
{
  Base* b= new Derived;
  delete b;
  return 0;

}
```

Output:

```
Base constructor is called
Derived class constructor is called
Base class object is destroyed
```

In the above example, delete b will only call the base class destructor due to which derived class destructor remains undestroyed. This leads to the memory leak.

- Example with a virtual destructor

```
#include <iostream>
using namespace std;
class Base
{
   public:
   Base()
   {
      cout<<"Base constructor is called"<<"\n";
```

```
    }
    virtual ~Base()
    {
        cout<<"Base class object is destroyed"<<"\n";
    }
};
class Derived:public Base
{
    public:
    Derived()
    {
        cout<<"Derived class constructor is called"<<"\n";
    }
    ~Derived()
    {
        cout<<"Derived class object is destroyed"<<"\n";
    }
};
int main()
{
  Base* b= new Derived;
  delete b;
  return 0;

}
```

Output:

```
Base constructor is called
Derived class constructor is called
Derived class object is destroyed
Base class object is destroyed
```

When we use the virtual destructor, then the derived class destructor is called first, and then the base class destructor is called.

Java Basics Interview Questions	**Java OOPs Interview Questions**
Java Multithreading Questions	**Java String & Exception Questions**
Java Collection Interview Questions	**JDBC Interview Questions**
Servlet Interview Questions	**JSP Interview Questions**
Spring Interview Questions	**Hibernate Interview Questions**
PL/SQL Interview Questions	**SQL Interview Questions**
Oracle Interview Questions	**Android Interview Questions**
SQL Server Interview Questions	**MySQL Interview Questions**

Heart Internet 24/7 support

Built by experts, powered by your imagination

heartinternet.uk

OPEN