

Short questions & answers Object Oriented Programming

1. What is OOP?

Stands for "Object-Oriented Programming." OOP (not Oops!) refers to a programming methodology based on objects, instead of just functions and procedures. These objects are organized into classes, which allow individual objects to be group together.

2. Write basic concepts of OOP?

Following are the concepts of OOP and are as follows:

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

3. What are the different features of c++?

Following are the different features of the Classes in C++,

- Operators and function overloading
- free storage management
- Constant types
- References
- Inline function
- Virtual function
- Templates
- Exception handling

4. What is class?

A class is simply a representation of a type of object. It is the template that describes the details of an object. A **class** can have subclasses that can inherit all or some of the characteristics of the **class**.

5. Write syntax of a class.

Syntax of Class declaration: A class definition starts with the keyword **class** followed by the class name. The class body, enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations.

Syntax:

```
class [class name]
{
access specifier:
statement(s);
};
```

Example: A class name "box".

```
class Box{
public:
double length; // Length of a box
double breadth; // Breadth of a box
double height; // Height of a box };
```

6. What is an object?

- Object is an instance of class.
- "object" refers to a particular instance of a class where the **object** can be a combination of variables, functions, and data structures.

7. Differentiate between class and object.

An object is an instance of a class. But classes don't have any information. Definition of properties and functions can be done at class and can be used by an object. Classes can have sub-classes but object doesn't have sub-objects.

8. What is Encapsulation?

Data encapsulation, sometimes referred to as data hiding, is the mechanism whereby the implementation details of a class are kept hidden from the user. The user can only perform a restricted set of operations on the hidden members of the class by executing special functions commonly called methods.

9. What is Polymorphism?

In object-oriented programming, polymorphism (from the Greek meaning "having multiple forms") is the characteristic of being able to assign a different meaning or usage to something in different contexts - specifically, to allow an entity such as a variable, a function, or an object to have more than one form.

10. What is dynamic or run-time polymorphism?

Dynamic or run-time polymorphism is also known as method overriding in which call to an overridden function is resolved during run-time, not at the compile time. It means having two or more methods with the same name, same signature but with different implementation.

11. Define Abstraction.

Abstraction is the one of the feature of OOP and it shows only the necessary details to the client of an object. Means, it shows only necessary details for an object, not the inner details of an object.

Example_ when you want to switch on television, it's not necessary to show all the functions of television .whatever is required to switch on TV will be showed by using abstract class.

12. What is Inheritance?

The process of creating new class from existed class is called inheritance. New classes retain characteristics of the existed class. A class from which new class is inherited is called Base Class. Derived class is called Child class.

Parent Class=Base Class=Super Class
Child Class=Derived Class

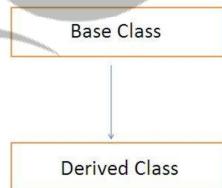
13. Enlist types of inheritance.

Types of Inheritance are as follows:

- 1- Single Inheritance.
- 2- Multiple Inheritance.
- 3- Multilevel Inheritance.
- 4- Hybrid Inheritance (also known as Virtual Inheritance)

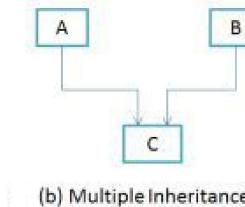
14. Define Single inheritance.

In C++ single inheritance only one class can be derived from the base class. Based on the visibility mode used or access specifier used while deriving, the properties of the base class are derived. Access specifier can be private, protected or public.



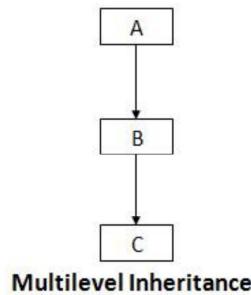
15. What is Multiple inheritance?

Multiple Inheritance in C++ Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes. The constructors of inherited classes are called in the same order in which they are inherited.



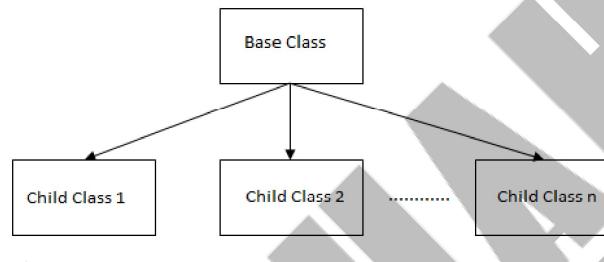
16. What is Multi-level inheritance?

When a class is derived from a class which is also derived from another class, i.e. a class having more than one parent classes, such inheritance is called Multilevel Inheritance.



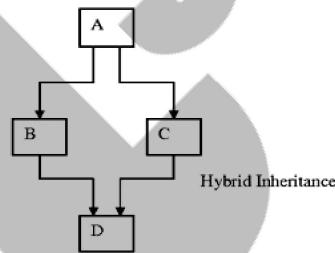
17. What is Hierarchical Inheritance?

When several classes are derived from common base class it is called hierarchical inheritance. In C++ hierarchical inheritance, the feature of the base class is inherited onto more than one sub-class.



18. What is Hybrid inheritance?

The inheritance in which the derivation of a class involves more than one form of any inheritance is called C++ hybrid inheritance.



19. What is Diamond problem?

In object-oriented programming languages with multiple inheritances, the diamond problem (sometimes referred to as the "deadly diamond of death") is an ambiguity that arises when two classes B and C inherit from A, and class D inherits from both B and C. If D calls a method defined in A (and does not override the method), and B and C have overridden that method differently, then from which class does it inherit: B, or C?

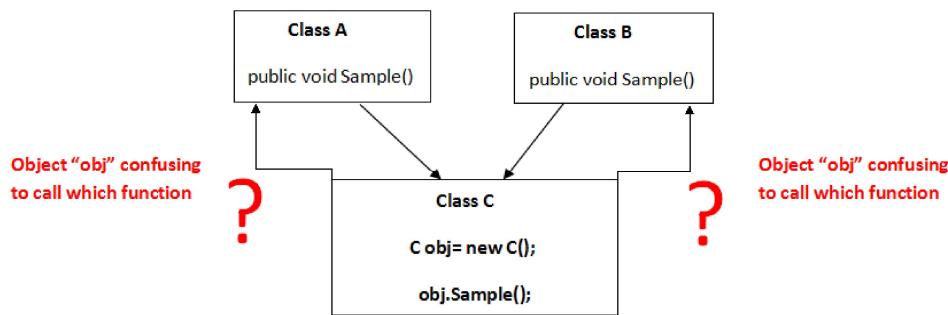


Figure 2: Diamond Problem

20. What is "this" pointer?

Every object in C++ has access to its own address through an important pointer called "this" pointer. The "this" pointer is an implicit parameter to all member functions. Therefore, inside a member function, this may be used to refer to the invoking object. Friend functions do not have a "this" pointer, because friends are not members of a class. Only member functions have a "this" pointer. Syntax:

```
return *this; OR this->member-identifier
```

21. What is wrapper class?

In general, a wrapper class is any class which "wraps" or "encapsulates" the functionality of another class or component. These are useful by providing a level of abstraction from the implementation of the underlying class or component.

22. Define a Constructor?

Constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules for constructor are: constructor name should be same as class name .constructor have no return type.

23. What are the various types of constructors?

They are as follows:

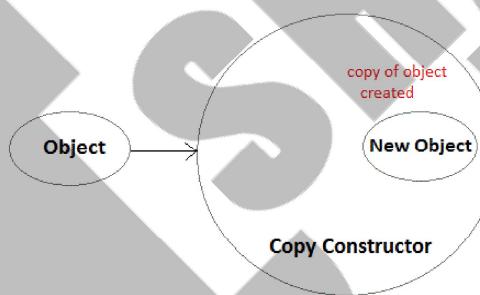
Default constructor- with no parameter

Parametric constructor- create a new instance of a class and also passing arguments simultaneously

Copy constructor- which creates new object as a copy of an existing object.

24. What is copy constructor?

The 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 **copy constructor** is used to: Initialize one object from another of the same type. **Copy** an object to pass it as an argument to a function.



25. What is parametric constructor?

When you define the **constructor's** body, use the parameters to initialize the object. When an object is declared in a **parameterized constructor**, the initial values have to be passed as arguments to the **constructor** function. The normal way of object declaration may not work.

26. What is default constructor on c++?

A default constructor is a constructor that either has no parameters, or if it has parameters, all the parameters have default values. If no user-defined constructor exists for a class A and one is needed, the compiler implicitly declares a default parameterless constructor A::A().

27. Why const keyword is used in copy constructor?

One reason for passing const reference is, we should use **const** in C++ wherever possible so that objects are not accidentally modified. ... So the compiler creates a temporary object which is copied to t2 using copy constructor in the original program (The temporary object is passed as an argument to copy constructor).

28. What is constructor overloading?

Constructor Overloading. Constructor can be overloaded in a similar way as function overloading. Overloaded constructors have the same name (name of the class) but different number of arguments. Depending upon the number and type of arguments passed, specific constructor is called.

29. What is Destructor?

Destructor functions are the inverse of constructor functions. They are called when objects are destroyed. Same name as class name with symbol (~) called tilde. It has no return type and there is only one destructor in one class.

30. What is inline function?

Inline function is a technique used by the compilers and instructs to insert complete body of the function wherever that function is used in the program source code.

31. What is exception handling?

Exception handling is the process of responding to the occurrence, during computation, of exceptions – anomalous or exceptional conditions requiring special processing – often changing the normal flow of program execution.

32. What is a ternary operator?

Ternary operator is said to be an operator which takes three arguments. Arguments and results are the different data types, and it depends on the function. Ternary operator is also called as conditional operator.

33. What are access specifiers?

These are used to determine the scope of the method or variables that can be accessed from other various objects or classes. There are following types of access specifiers:

- Private
- Protected
- Public

34. Define super keyword?

It is used to invoke overridden method which overrides one of its super class methods. This keyword allows to access overridden methods and also to access hidden members of super class.

35. What is operator overloading? Can we introduce a new operator by using operator overloading?

Operator overloading is an important concept in C++. It is a type of polymorphism in which an operator is overloaded to give user defined meaning to it. Overloaded operator is used to perform operation on user-defined data type. For example '+' operator can be overloaded to perform addition on various data types, like for Integer, String. No, we can only overload the operators that exist. We cannot create new operators or rename existing operators. For example, we could not create an operator ** to do exponents.

36. Which operator cannot be overloaded?

Following operators cannot be overloaded:

- Scope resolution operator (::)
- Member selection (.)

37. What is operator overloading?

Operator overloading is a function where different operators are applied and depends on the arguments. Operator -, *, = can be used to pass through the function, and it has their own precedence to execute.

38. What is operator overriding?

39. What is function overloading?

Function overloading is defined as a normal function, but it has the ability to perform different tasks. It allows creation of several methods with the same name which differ from each other by type of input and output of the function.

Example:

```
Void add(int a, int b)
Void add(double a, double b)
```

40. What is function overriding?

It is the feature that allows sub class to provide implementation of a method that is already defined in the main class. This will overrides the implementation in the super class by providing the same method name, same parameter and same return type.

41. What is the difference between function overloading and function overriding?

Function Overriding: Function Overriding also allows to have two or more function with the same name but in this case, it is important to have Inheritance. When a function is already defined in the base class and we redefine it in derived class, that function is said to be overridden. In other words, if an inherited class contains a function with same name as that of its parent, then this concept is known as Function Overriding.

Function Overloading: Function overloading allows us to define two or more function having the same name but with different function body. It is useful in real world applications because practically, we encounter such situations very commonly.

42. What is virtual function?

A virtual function is a member function of class and its functionality can be overridden in its derived class. This function can be implemented by using a keyword called virtual, and it can be given during function declaration. Virtual function can be achieved in c++, and it can be achieved in c language by using function pointers or pointers of function.

43. What is pure virtual function?

A pure virtual function is a function which can be overridden in the derived class but cannot be defined. A virtual function can be declared as pure by using the operator =0. Example:

```
Virtual void function()           // virtual not pure  
Virtual void function2()=0        // pure virtual
```

44. What is abstract class?

An abstract class is a class which cannot be initiated. Creation of an object is not possible in an object class, but it can be inherited. Your any class has at least one method abstract than your class is abstract class.

45. What is friend function?

Friend function is a friend of class that is allowed to access to public, private or protected data in the same class. If the function is defined outside the class cannot access such information. Friend can be declared anywhere in the class declaration, and it cannot be affected by access control keywords like private, public or protected.



Visit

<https://tshahab.blogspot.com>

for more.