Interview Question 1

What do the letters in OOP signify?

Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, Encapsulation etc.

What is a class?

It is similar to structures in C language. Class can also be defined as user defined data type but it also contains functions in it. So, class is basically a blueprint for an object. It declares & defines what data variables the object will have and what operations can be performed on the class's object.

```
class A

{

   public:

        static int integer;

        int getInt();

        //A();

};
```

What is an object?

Objects are the basic unit of OOP. They are instances of class, which have data members and use various member functions to perform tasks.

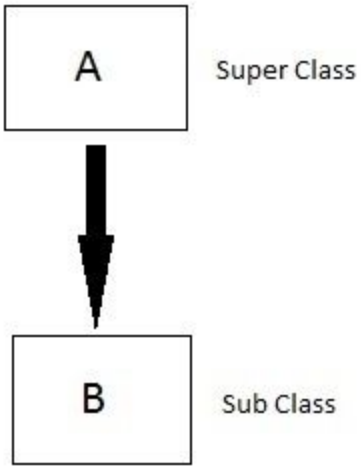
```
A firstInstance;

A secondInstance;
```

What are the 5 types of inheritance in C++?

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

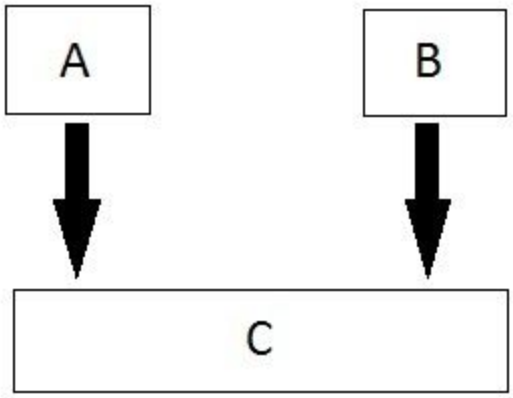
Single Inheritance in C++

In this type of inheritance one derived class inherits from only one base class. It is the simplest form of Inheritance.



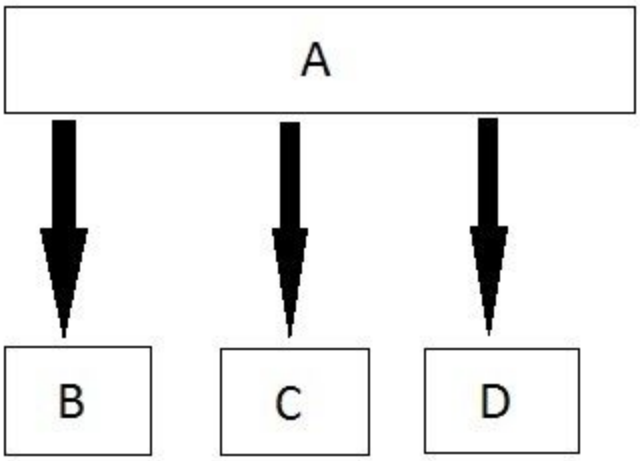
Multiple Inheritance in C++

In this type of inheritance a single derived class may inherit from two or more than two base classes.



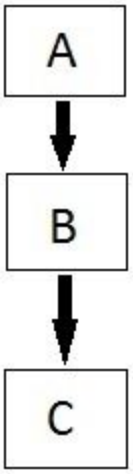
Hierarchical Inheritance in C++

In this type of inheritance, multiple derived classes inherit from a single base class.



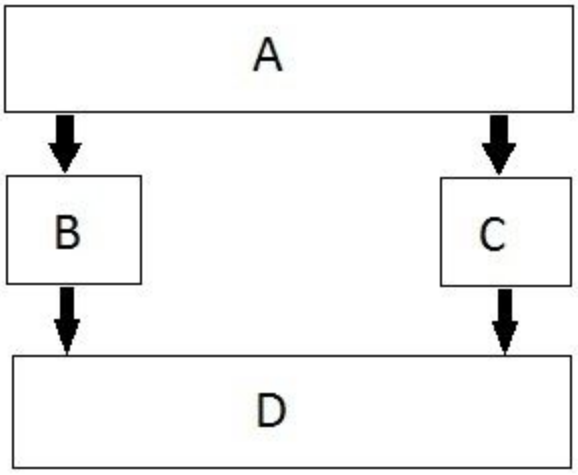
Multilevel Inheritance in C++

In this type of inheritance the derived class inherits from a class, which in turn inherits from some other class. The Super class for one, is sub class for the other.



Hybrid (Virtual) Inheritance in C++

Hybrid Inheritance is a combination of Hierarchical and Multilevel Inheritance.



What are the three access class specifiers in C++?

Access specifiers can be either private or protected or public. In general access specifiers are the access restriction imposed during the derivation of different subclasses from the base class.

- private access specifier
- protected access specifier
- public access specifier

What is encapsulation?

It can also be said to be data binding. Encapsulation is all about binding the data variables and functions together in class.

For example: getter, setter.

What is data abstraction?

Abstraction refers to showing only the essential features of the application and hiding the details. In C++, classes can provide methods to the outside world to access & use the data variables, keeping the variables hidden from direct access, or classes can even declare everything accessible to everyone, or maybe just to the classes inheriting it. This can be done using access specifiers.

What is inheritance?

Inheritance is a way to reuse once written code again and again. The class which is inherited is called the **Base** class & the class which inherits is called the **Derived** class. They are also called parent and child class.

So when a derived class inherits a base class, the derived class can use all the functions which are defined in the base class, hence making code reusable.

What is polymorphism?

It is a feature, which lets us create functions with the same name but different arguments, which will perform different actions. That means, functions with the same name, but functioning in different ways. Or, it also allows us to redefine a function to provide it with a completely new definition. You will learn how to do this in detail soon in coming lessons.

What is an advantage of inline functions?

Function call overhead doesn't occur.

It also saves the overhead of push/pop variables on the stack when the function is called.

It also saves overhead of a return call from a function.

When you inline a function, you may enable the compiler to perform context specific optimization on the body of function. Such optimizations are not possible for normal function calls. Other optimizations can be obtained by considering the flows of calling context and the called context.

Inline function may be useful (if it is small) for embedded systems because inline can yield less code than the function call preamble and return.

What makes a class an abstract class?

A pure virtual function (or abstract function) in C++ is a virtual function for which we don't have implementation, we only declare it. A pure virtual function is declared by assigning 0 in declaration. See the following example.

```
class Base {

   virtual void method() {std::cout << "from Base" <<
std::endl;}

public:

   virtual ~Base() {method();}

   void baseMethod() {method();}

};

class A : public Base {

   void method() {std::cout << "from A" << std::endl;}

public:

   ~A() {method();}

};
```

Summary

- ❏ OOP for reusability, extensibility, maintainability
- ❏ A class is blueprint to create objects
- ❏ Object is an instance of the class
- ❏ A class has attributes and methods: data & Functions
- ❏ Members can be private, protected or public.
- ❏ This pointer is passed to methods
- ❏ New to create (Constructor)

- ❏ Delete to destroy (Destructor)
- ❏ UML diagram for class
- ❏ Encapsulation is all about binding the data variables and functions
- ❏ Abstraction refers to showing only the essential features of the application and hiding the details.
- ❏ Inheritance is a way to reuse once written code again and again.
- ❏ Polymorphism is a feature, which lets us create functions with the same name but different arguments, which will perform different actions.

Reference

Slide OOP slide Tran Duy Quang 17CTT2

OOP concept in C++: https://www.studytonight.com/cpp/cpp-and-oops-concepts.php

http://www.tutorialdost.com/Cpp-Programming-Tutorial/49-Cpp-Types-of-Inheritance.aspx

Code Demo: https://www.geeksforgeeks.org/inheritance-in-c/