

C++ NOTE 4: Inheritance

By now we have worked with Constructors and destructors and also have understood some basic features or functionalities of OOP design technique.

Inheritance:

We have the preliminary knowledge of what Inheritance is, what are based and derived classes and also the different types of inheritance present in C++.

Now we go into detail to understand the concept of Inheritance in depth and more practically.

Publicly, Privately and Protectedly Derived Classes:

Before we go into Multi level and Multiple inheritance we need to first understand the different ways in which we can derive classes, namely: Public derivation, Protected derivation and Private derivation.

In all the programs that we wrote upto now that were based on the concept of inheritance we publicly derived a base class into an derived class. This derivation is the most commonly used and will be generally used in your exams. But what does it mean to "Publicly Inherit" a class?

We understand that a class can have various data members and member functions which can be written under either private, public or protected(only used in special cases of Inheritance).

The private members cannot be directly accessed from outside the class that is no object of the class can directly access members but can do so through a public member function. Recall an example we did in the previous classes.

So if I inherit that particular class into another class, should the private members be inherited too? No, these are private to that class and cannot be inherited.

For eg: https://drive.google.com/file/d/1z88Tjz7Ti9ecTqi_ILmxcrVZWVggzSxZ/view?usp=sharing

So as we see in the example the private members are not inherited and hence cannot be accessed through the public member function in the derived class. But if we want a data member that acts as private but can be inherited to other derived classes we use the "Protected keyword".

For eg: https://drive.google.com/file/d/1lm8I93kznXG3GU7VB5PSLAE0RCO_U2o/view?usp=sharing

Here we see that the public member functions of the derived class can access the protected data members of the base class. This happens because the protected members are inherited to the derived class as protected members.

To get a general Idea about what derivation does what, follow this table:

Base class visibility	Derived class visibility		
	Public	Private	Protected
Private	Not Inherited	Not Inherited	Not Inherited
Protected	Protected	Private	Protected
Public	Public	Private	Protected

Types of Inheritance:

Here are some examples to show the function of different types of inheritance in C++:

1. Single Level Inheritance:
https://drive.google.com/file/d/1q51LY4oK1i0UOp25KCmUSjTE1A6Pr_wO/view?usp=sharing
2. Multilevel Inheritance:
https://drive.google.com/file/d/1OyReI5NZSww7Zt3C8oRc31_WxRX0qt41/view?usp=sharing
3. Multiple Inheritance:
<https://drive.google.com/file/d/15-3nZ5GHMAHktFN87t1kcg9ZAnezxcef/view?usp=sharing>