# C++ Inheritance

### 

### What is Inheritance in C++?

The concept of reusability in C++ is supported using inheritance. We can reuse the properties of an existing class by inheriting it and deriving its properties. The existing class is called the base class and the new class which is inherited from the base class is called the derived class.

The syntax for inheriting a class is

// Derived Class syntax

class derived\_class\_name : access\_modifier base\_class\_name

{

// body of the derived class

}

### Types of inheritance in C++

#### Single Inheritance

Single inheritance is a type of inheritance in which a derived class is inherited with only one base class.

For example, we have two classes ClassA and ClassB.  If ClassB is inherited from ClassA, it means that ClassB can now implement the functionalities of ClassA. This is single inheritance.

class ClassA

{

//body of ClassA

};

//derived from ClassA

class ClassB : public ClassA

{

//body of ClassB

};

#### Multiple Inheritance

Multiple inheritances is a type of inheritance in which one derived class is inherited from more than one base class.

For example, we have three classes ClassA, ClassB, and ClassC.  If ClassC is inherited from both ClassA & ClassB, it means that ClassC can now implement the functionalities of both ClassA & ClassB. This is multiple inheritances.

class ClassA

{

//body of ClassA

};

class ClassB

{

//body of ClassB

};

//derived from ClassB and Class C

class ClassC : public ClassA, public ClassB

{

//body of ClassC

};

* **Hierarchical Inheritance**

A hierarchical inheritance is a type of inheritance in which several derived classes are inherited from a single base class.

For example, we have three classes ClassA, ClassB, and ClassC.  If ClassB and Class C are inherited from ClassA, it means that ClassB and ClassC can now implement the functionalities of ClassA. This is hierarchical inheritance.

class ClassA

{

//body of ClassA

};

//derived from ClassA

class ClassB : public ClassA

{

//body of ClassB

};

//derived from ClassA

class ClassC : public ClassA

{

//body of ClassC

};

* **Multilevel Inheritance**

Multilevel inheritance is a type of inheritance in which one derived class is inherited from another derived class.

For example, we have three classes ClassA, ClassB, and ClassC.  If ClassB is inherited from ClassA and ClassC is inherited from ClassB, it means that ClassB can now implement the functionalities of ClassA and ClassC can now implement the functionalities of ClassB. This is multilevel inheritance.

class ClassA

{

//body of ClassA

};

//derived from ClassA

class ClassB : public ClassA

{

//body of ClassB

};

//derived from ClassB

class ClassC : public ClassB

{

//body of ClassC

};

* **Hybrid Inheritance**

Hybrid inheritance is a combination of different types of inheritances.

For example, we have four classes ClassA, ClassB, ClassC, and ClassD.  If ClassC is inherited from both ClassA and ClassB and ClassD is inherited from ClassC, it means that ClassC can now implement the functionalities of both ClassA and ClassB and ClassD can now implement the functionalities of ClassC. This is multilevel inheritance where both multilevel and multiple inheritances are present.

class ClassA

{

//body of ClassA

};

class ClassB

{

//body of ClassB

};

//derived from ClassA and ClassA

class ClassC : public ClassA, public ClassB

{

//body of ClassC

};

//derived from ClassC

class ClassD : public ClassC

{

//body of ClassD

};