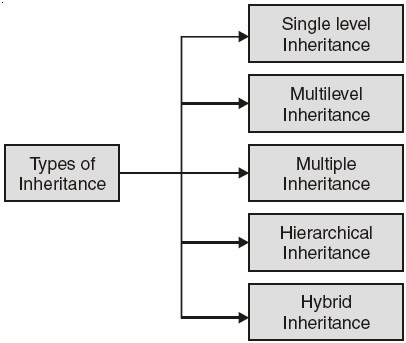
**9.2 Types of Inheritance**

In general inheritance is of five types:

1. Single level inheritance
2. Multilevel inheritance
3. Multiple inheritances
4. Hierarchical inheritance
5. Hybrid Inheritance

  
**Figure 9.1:**Different types of inheritance

Again depending upon in which mode we do the above any 5 of the inheritance we can further divide inheritance as :

1. Public Inheritance
2. Private Inheritance
3. Protected Inheritance

The syntax of deriving a new class from an already existing class is shown as :

class new\_class\_name : mode old\_class\_name

{

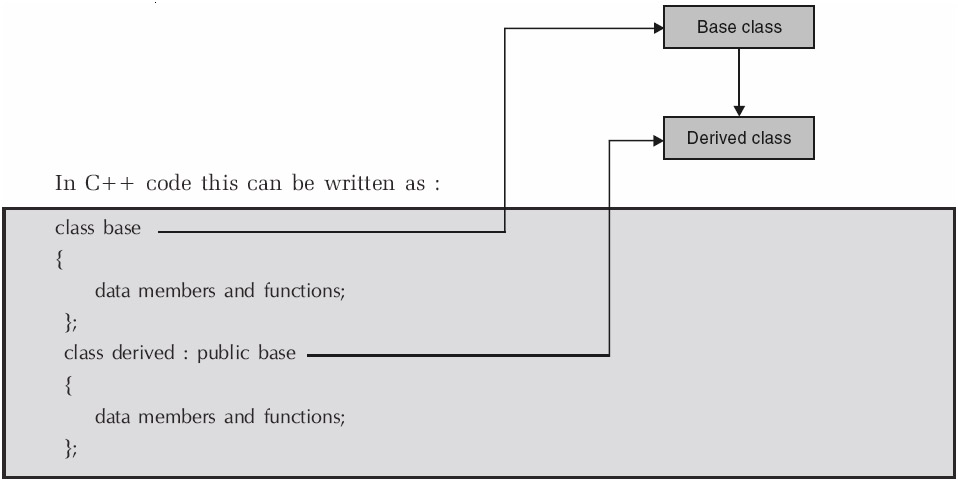
};

Where class is keyword used to create a class **new\_class\_name** of new derived class, mode may be **private**, **public**, **or protected** or even be absent *i.e.*, be an optional. If mode is not present default mode private is assumed. **Old\_class\_name** is the name of an already existing class. It may be a user defined or a built-in class.

1. **Single Level Inheritance**

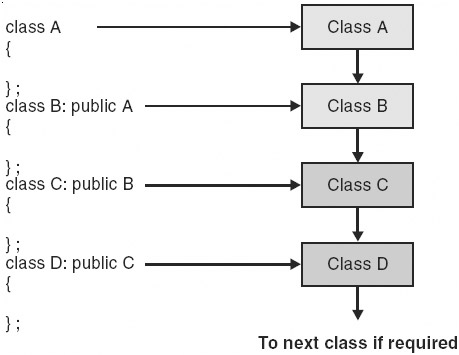
In single level inheritance we have just one base class and one derived class. It is represented as :

In C++ code this can be written as :



1. **Multilevel Inheritance**

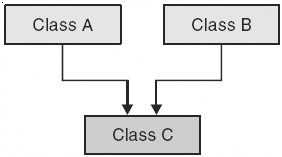
In multiple inheritances we have one base class and one derived at one level. At the next level the derived class becomes base class for the next class and so on. This is as shown on page 395.



The class A and B together forms one level, class B and class C together forms another level and so on. For a class B, class A is the parent and for class C, class B is the present thus in this inheritance level we can say that A is the grandfather of class C and class C is the grandchild of class A.

1. **Multiple Inheritance**

In multiple inheritance a child can have more than parent i.e., a child can inherit properties from more than one class. Diagrammatically this is as shown below :



In C++ code for the diagram is given as follows :

class A class B

{ {

}; };

class C: public A, public B

{

};

The mode need not be the same. The **class A** may be inherited in **public** and **class B** in **private** or whatever mode as per the user desired. Note mode has to be specified for both the classes. If you write as :

class C: public A, B

{

};

Than it does not mean **both A** and **B** are inherited in **class C** in **public** mode. The class **A** is inherited in **public** and class **B** in private which is the **default** mode. Again if you write.

class C : A,B

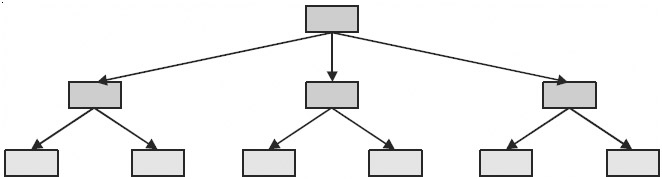
{

};

Than both classes A and B are inherited to C in **private** mode.

1. **Hierarchical Inheritance**

In this type of inheritance multiple classes share the same base class. That is number of classes inherits the properties of one common base class. The derived classes again may become base class for other classes. This is shown as follows :

  
**Figure 9.2:**Show the hierarchical inheritance

For example, a university has number of colleges under its affiliation. Each college may use the university name, the chairperson name, its address, phone number etc.

There are number of properties or features which a vehicle posses. The common properties of all the vehicle may be put under one class vehicle and different classes like two-wheeler, four-wheeler and three-wheeler can inherit the vehicle class.

As another example in an engineering college various departments be termed as various classes which may have one parents class common, the name of engineering college. Again for each department there may be various classes like Lab\_staff, Faculty class etc. In C++ code the first level can be seen as follows :

class A

{

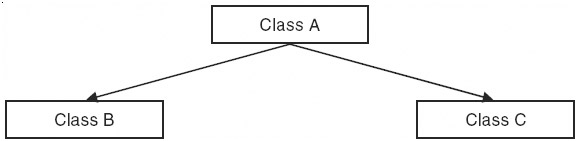
};

class B : public A class C : public A class D : public A

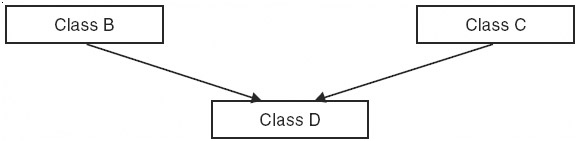
{ }; { }; { };

1. **Hybrid Inheritance**
   1. Let's consider the figure as shown below :

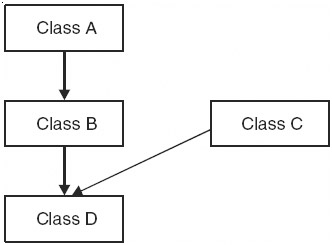
For the first half of the figure, we have hierarchical inheritance as shown by breaking the figure as :

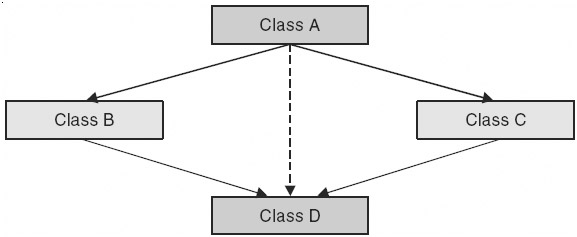


In second half we have multiple inheritance as shown in the figure :



1. The second figure for hybrid inheritance may be viewed as :





**Figure 9.3:**Implementation of hybrid inheritance

The C++ code may be written as follows :

class B :public A

{ };

class D :public B, public C

{

};

The inheritance is hybrid as it involves multilevel and multiple inheritance.