Department Of Commerce (CA)

Object Oriented programming with C++

Sem: II Subcode: 18MCC23C

Unit-IV I M.Com(CA)

Operator overloading - overloadable operators - Rules for overloading operators - Inheritance - Forms of inheritance - Single, Multiple, Multilevel, hierarchal and hybrid inheritance - When to use inheritance - Benefits of inheritance.

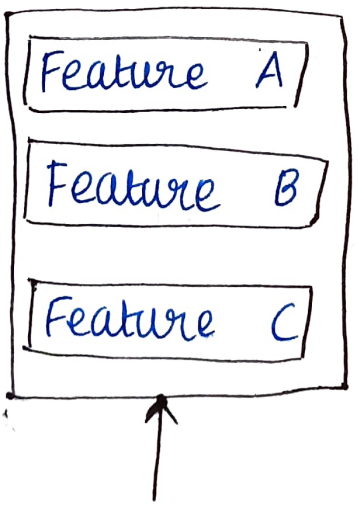
Reference Books :-

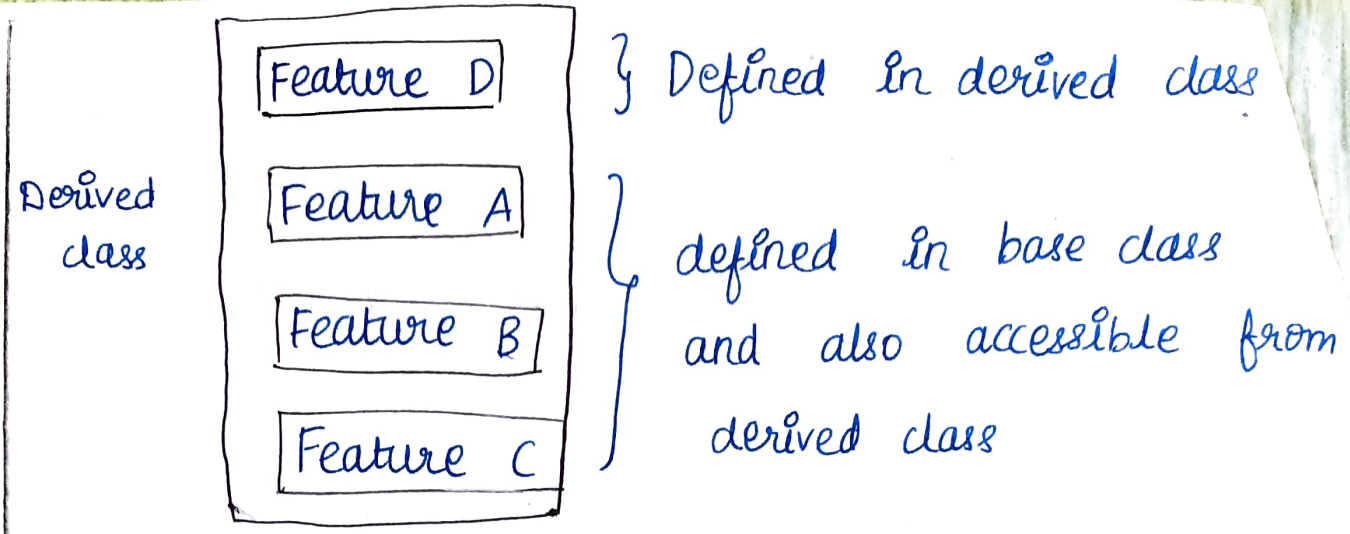
1. Programming with C++ - D. Ravichandran
2. Mastering C++ - K.R. Venugopal, Rajkumar T. Ravishanker.
3. Object Oriented Programming with C++ - E. Balagurusamy.

Prepared By
Dr. S. VASANTHA
Assist Professor.

INHERITANCE

Inheritance allows new classes to be built from older and less specialized classes instead of being rewritten from scratch. Classes are created by first inheriting all the variables and behaviour defined by some primitive class and then adding specialized variables and behaviour. In object oriented programming, classes encapsulate data and functions into one package. New classes can be built from existing ones, just as builder constructs a skycraper out of bricks, stone and other relatively simple material. The technique of building new classes from the existing classes is called inheritance.

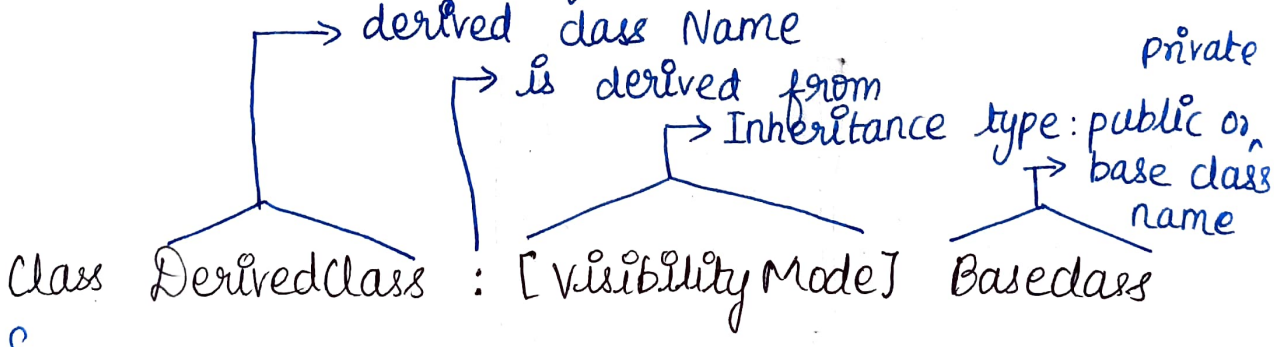




Base class and derived class relationship

Derived Class Declaration:

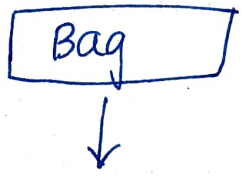
A derived class extends its features by inheriting the properties of another class, called base class and adding features of its own. The declaration of a derived class specifies its relationship with the base class in addition to its own features.




```
{
    // members of derived class .
    // and they can access members of the base class
};
```

A Sample Program on Single Inheritance:

A derived class may begin its existence with a copy of its base class members, including any other members inherited from more distantly related classes. A derived class inherits data members and member functions, but not the constructor or destructor from its base class.

class Bag 

Base class

class Set : Public Bag 

Derived class

Forms of Inheritance:

Inheritance is classified into the following forms based on the levels of inheritance and interrelation among the classes involved in the inheritance process:

(i) Single Inheritance: Derivation of a class from only one base class is called single Inheritance. The sample program, union.cpp, discussed above falls under this category.

(ii) Multiple Inheritance: Derivation of a class f... several base classes is called Multiple Inherita...

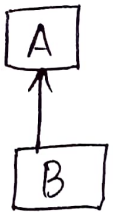
(iii) Hierarchical Inheritance: Derivation of several cla... from a single base class, i.e., the traits of one class may be inherited by more than one class, is called Hierarchical Inheritance.

Multilevel Inheritance: Derivation of a class from another derived class is called Multilevel Inheritance.

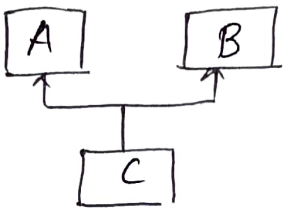
Hybrid Inheritance: Derivation of a class involving more than one form of inheritance is known as hybrid Inheritance.

Multipath Inheritance: Derivation of a class from other derived classes, which are derived from the same base class is called multipath Inheritance.

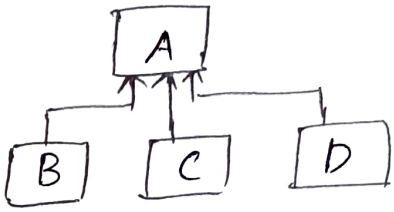
a) Single Inheritance



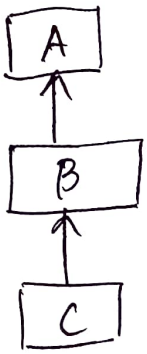
b) Multiple Inheritance



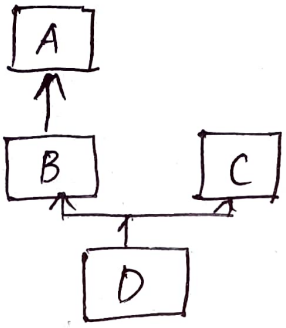
c) Hierarchical Inheritance



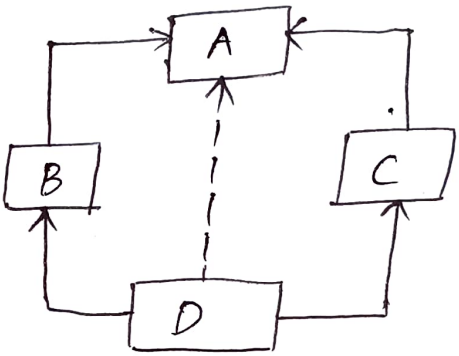
d) Multilevel Inheritance



e) Hybrid Inheritance



f) Multipath Inheritance



Example:

```
# include <iostream.h>
class B          // base class
{
    // body of base class, without constructors
};
class D : public B    // publicly derived class
{
    // body of derived base class, without constructors
    public :
    void msg()
```

```
{
    cout << " No constructors exists in base and
            derived class " << endl ;
}
};
void main ()
{
    D objd; // base constructor
    objd.msg();
}
```