INHERITANCE

- **Inheritance is the process by which new classes called *derived* classes are created from existing classes called *base* classes**.
- The derived classes have all the features of the base class and the programmer can choose to add new features specific to the newly created derived class.
- The idea of inheritance implements the **is a** relationship. For example, mammal IS-A animal, dog IS-A mammal hence dog IS-A animal as well and so on.

Features or Advantages of Inheritance:

- ***Reusability of Code***
- ***Saves Time and Effort***
- ***Faster development, easier maintenance and easy to extend***
- ***Capable of expressing the inheritance relationship and its transitive nature which ensures closeness with real world problems .***

Base & Derived Classes:

A class can be derived from more than one classes, which means it can inherit data and functions from multiple base classes. A class derivation list names one or more base classes and has the form:

class derived-class: access-specifier base-class

Where access is one of **public, protected,** or **private**.

For example, if the *base* class is *MyClass* and the derived class is sample it is specified as:

class sample: public MyClass

The above makes sample have access to both *public* and *protected* variables of base class *MyClass*.

EXAMPLE OF SINGLE INHERITANCE

Consider a base class **Shape** and its derived class **Rectangle** as follows:

```
// Base class
class Shape
{
  public:
    void setWidth(int w)
    {
      width = w;
    }
    void setHeight(int h)
    {
      height = h;
    }
  protected:
    int width;
    int height;
};
```

```
// Derived class
class Rectangle: public Shape
{
  public:
    int getArea()
    {
      return (width * height);
    }
};
int main(void)
{
  Rectangle Rect;
   Rect.setWidth(5);
  Rect.setHeight(7);
  // Print the area of the object.
  cout << "Total area: " << Rect.getArea() << endl;

  return 0;

}
```

When the above code is compiled and executed, it produces following result:

Total area: 35

Access Control and Inheritance:

A derived class can access all the non-private members of its base class. Thus base-class members that should not be accessible to the member functions of derived classes should be declared private in the base class.We can summarize the different access types according to who can access them in the following way:

Access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

A derived class inherits all base class methods with the following exceptions:

- Constructors, destructors and copy constructors of the base class.
- Overloaded operators of the base class.
- The friend functions of the base class.

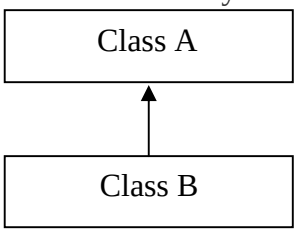
When deriving a class from a base class, the base class may be inherited through **public, protected** or **private** inheritance. We hardly use **protected** or **private** inheritance but **public** inheritance is commonly used. While using different type of inheritance, following rules are applied:

1. **Public Inheritance:** When deriving a class from a **public** base class, **public** members of the base class become **public** members of the derived class and **protected** members of the base class become **protected** members of the derived class. A base class's **private** members are never accessible directly from a derived class, but can be accessed through calls to the **public** and **protected** members of the base class.
2. **Protected Inheritance:** When deriving from a **protected** base class, **public** and **protected** members of the base class become **protected** members of the derived class.
3. **Private Inheritance:** When deriving from a **private** base class, **public** and **protected** members of the base class become **private** members of the derived Class.

TYPES OF INHERITANCE

1. Single class Inheritance:

- Single inheritance is the one where you have a single base class and a single derived class.

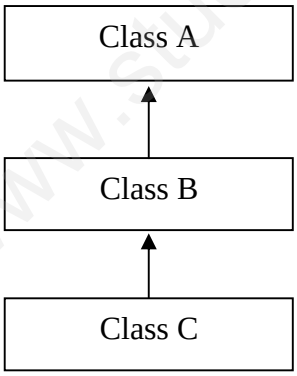


a Base class (super)

it is a sub class (derived)

2. Multilevel Inheritance:

- In Multi level inheritance, a subclass inherits from a class that itself inherits from another class.



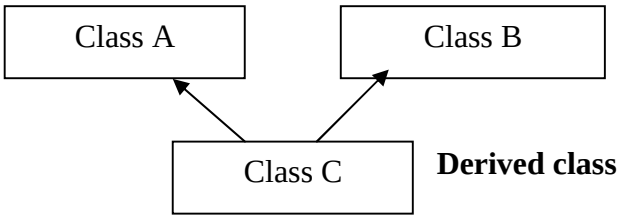
it is a Base class (super) of B

it is a sub class (derived) of A
and base class of class C

derived class(sub) of class B

3. Multiple Inheritance:

- In Multiple inheritances, a derived class inherits from multiple base classes. It has properties of both the base classes.

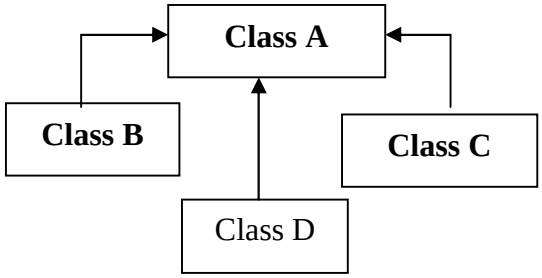


Base class

Derived class

4. Hierarchical Inheritance:

- In hierarchial Inheritance, it's like an inverted tree. So multiple classes inherit from a single base class.

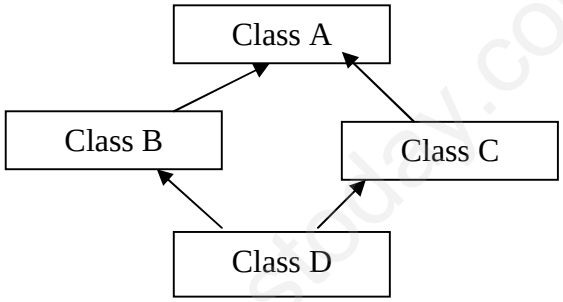


Base Class

Sub Class(derived)

5. Hybrid Inheritance:

- It combines two or more forms of inheritance .In this type of inheritance, we can have mixture of number of inheritances but this can generate an error of using same name function from no of classes, which will bother the compiler to how to use the functions.
- Therefore, it will generate errors in the program. This has known as ambiguity or duplicity.
- Ambiguity problem can be solved by using **virtual base classes**



4 marks Solved Problems :

Q1. **Consider the following declarations and answer the questions given below :**

```
class WORLD
{
int H;
protected :
int S;
public :
void INPUT(int);
void OUTPUT();
};
class COUNTRY : private WORLD
{
int T;
protected :
int U;
public :
void INDATA( int, int)
void OUTDATA();
};
class STATE : public COUNTRY
{
int M;
public :
void DISPLAY (void);};
```

(*i*) Name the base class and derived class of the class COUNTRY.
(*ii*) Name the data member(s) that can be accessed from function DISPLAY().
(*iii*) Name the member function(s), which can be accessed from the objects of class STATE.
(*iv*) Is the member function OUTPUT() accessible by the objects of the class COUNTRY ?
Ans (*i*) Base class : WORLD
Derived class : STATE
(*ii*) M.
(*iii*) DISPLAY(), INDATA() and OUTDATA()
(*iv*) No

Q2. Consider the following declarations and answer the questions given below :

```
class living_being {
char name[20];
protected:
int jaws;
public:
void inputdata(char, int);
void outputdata();
}
class animal : protected living_being {
int tail;
protected:
int legs;
public:
void readdata(int, int);
void writedata();
};
class cow : private animal {
char horn_size;
public:
void fetchdata(char);
void displaydata();
};
```

(*i*) Name the base class and derived class of the class animal.
(*ii*) Name the data member(s) that can be accessed from function displaydata.
(*iii*) Name the data member(s) that can be accessed by an object of cow class.
(*iv*) Is the member function outputdata accessible to the objects of animal class.
Ans (*i*) Base class : living_being
Derived class : cow
(*ii*) horn_size, legs, jaws
(*iii*) fetchdata() and displaydata()
(*iv*) No

Q3. **Consider the following and answer the questions given below :**

```
class MNC
{
char Cname[25]; // Company name
protected :
char Hoffice[25]; // Head office
public :
MNC( );
char Country[25];
void EnterDate( );
void DisplayData( );
};
```

```
class Branch : public MNC
{
long NOE; // Number of employees
char Ctry[25]; // Country
protected:
void Association( );
public :
Branch( );
void Add( );
void Show( );
};
class Outlet : public Branch
{
char State[25];
public :
Outlet();
void Enter();
void Output();};
```

(*i*) Which class's constructor will be called first at the time of declaration of an object of class Outlet?
(*ii*) How many bytes an object belonging to class Outlet require ?
(*iii*) Name the member function(s), which are accessed from the object(s) of class Outlet.
(*iv*) Name the data member(s), which are accessible from the object(s) of class Branch.

Ans (*i*) class MNC
(*ii*) 129
(*iii*) void Enter(), void Output(), void Add(), void Show(), void EnterData(), void DisplayData().
(*iv*) char country[25]

Q4 Consider the following and answer the questions given below :

```
class CEO {
double Turnover;
protected :
int Noofcomp;
public :
CEO( );
void INPUT( );
void OUTPUT( );
};
class Director : public CEO {
int Noofemp;
public :
Director( );
void INDATA();
void OUTDATA( );
protected:
float Funda;
};
class Manager : public Director {
float Expense;
public :
Manager();
void DISPLAY(void);
};
```

(*i*) Which constructor will be called first at the time of declaration of an object of class Manager?
(*ii*) How many bytes will an object belonging to class Manager require ?
(*iii*) Name the member function(s), which are directly accessible from the object(s) of class Manager.
(*iv*) Is the member function OUTPUT() accessible by the objects of the class Director ?
Ans (*i*) CEO()
(*ii*) 16
(*iii*) DISPLAY(), INDATA(), OUTDATA(), INPUT(), OUTPUT()
(*iv*) Yes

4 marks Practice Problems:

Q1 :- Consider the following declarations and answer the questions given below:

```
class vehicle
        {int wheels;
        protected:
        int passenger;
        public:
        void inputdata( int, int);
        void outputdata();};
        class heavyvehicle: protected vehicle
        {int dieselpetrol;
        protected:
        int load;
        public:
        void readdata( int, int);
        void writedata();};
        class bus:private heavyvehicle
        {char marks[20];
        public:
        void fetchdata(char);
        void displaydata();};
```

(i) **Name the class and derived class of the class heavyvehicle.**
(ii) **Name the data members that can be accessed from function displaydata()**
(iii) **Name the data members that can be accessed by an object of bus class**
(iv) **Is the member function outputdata() accessible to the objects of heavyvehicle class.**

Q2:- Consider the following declarations and answer the questions given below:

```
        class book
        {
        char title[20];
        char author[20];
        int noof pages;
        public:
                void read();
                void show();};
        class textbook: private textbook
        {int noofchapters, noofassignments;
        protected:
        int standard;
        void readtextbook();
        void showtextbook();};
        class physicsbook: public textbook
        {char topic[20];
```

```
public:
void readphysicsbook();
void showphysicsbook();}
```

(i) Name the members, which can be accessed from the member functions of class physicsbook.
(ii) Name the members, which can be accessed by an object of Class textbook.
(iii) Name the members, which can be accessed by an object of Class physicsbook.
(iv) What will be the size of an object (in bytes) of class physicsbook.

Q3 : Answer the questions (i) to (iv) based on the following:

```
class CUSTOMER
{       int Cust_no;
        char Cust_Name[20];
        protected:
        void Register();
public:
        CUSTOMER( );
        void Status( );};
class SALESMAN
{       int Salesman_no;
        char Salesman_Name[20];
        protected:
          float Salary;
        public:
        SALESMAN( );
        void Enter( );
        void Show( );};
class SHOP :  private  CUSTOMER, public SALESMAN
{               char Voucher_No[10];
                char Sales_Date[8;
        public :
                SHOP( );
                void  Sales_Entry( );
                void  Sales_Detail( );};
```

(i) Write the names of data members, which are accessible from object belonging to class CUSTOMER.
(ii) Write the names of all the member functions which are accessible from object belonging to class SALESMAN.
(iii) Write the names of all the members which are accessible from member functions of class SHOP.
(iv) How many bytes will be required by an object belonging to class SHOP?

2marks Practice Problems:

1. What is access specifier ? What is its role ?
2. What are the types of inheritance ?
3. What is the significance of inheritance ?
4. What is the difference between private and public visibility modes?