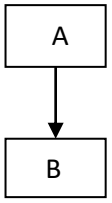
Inheritance

Inheritance is another key features of object-oriented programming. It is a form of software reusability in which new classes are created from existing classes by absorbing their attributes and behaviors ie properties and methods. Software reusability saves time in program development. It encourages reuse of proven and debugged high quality software, thus reducing problems after a system becomes operational.

There are many types of inheritance…….

(i) Single inheritance
(ii) Multiple inheritance (Interface)
(iii) Multilevel Inheritance
(iv) Hierarchical Inheritance

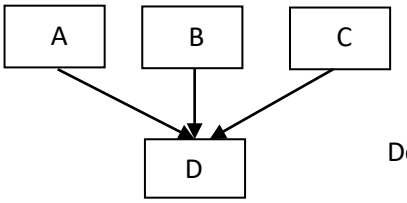
(I) **Single inheritance:-** It is the process of deriving a sub class from only one super class (base class) is called single inheritance.



Base Class(Super)

Derived class

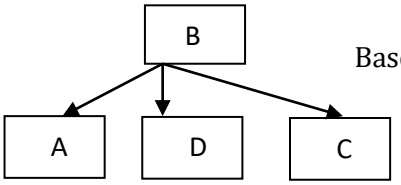
(II) **Multiple Inheritance** :-- It is the process of deriving a sub Class from many super classes called multiple inheritance.



Base Class(Super)

Derived class(Sub Class)

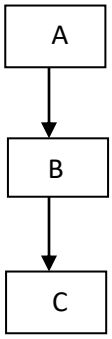
(III) **Hierarchical Inheritance** :-- The super class has many sub class is called hierarchical inheritance.



Base Class(Super)

Derived class(Sub Class)

(v) **Multilevel Inheritance** :-- It is the process of deriving a sub class from another sub class is called multilevel inheritance.



Sub Class

extends and super keyword:- When creating a new class, instead of writing completely new instance variables and instance methods, the programmer can designate that the new class is to inherit the instance variables and instance methods of a previously defined super class. The new class is referred to as subclass. Inheritance is done by using extends keyword

Syntax :-- Class subclass_name extends Super class_name

```
{
    Variable declaration;
    Methods declaration;
}
```

The super keyword is used to call the constructor of superclass. The call to the superclass constructor must be the first line in the body of the subclass constructor. It is also a syntax error if the arguments to a super call by a subclass to its superclass constructor do not match the parameters specified in one of the superclass constructor definitions

.

e.g. :--

```
class    Add
 {
        int x,y;
        Add( int a, int b)
       {
          X=a;
          Y=b;
       }
        int  addition ()
        {
              return ( x+y);
         }
    }
class ex_ add extends Add
 {
    int z;
    ex_ add (int a, int b, int c)
    {
        Super (a,b);
         Z=c;
     }
       int ex_ addition ()
   {
        return (x+y+z);
    }
```

```
        }


        class main_in
          {
            public static void main ( String agrs[])
            {
              ex_add a1 = new ex_ add (10,20,30);
               int sum1 = a1.add ();
              int sum2 = a1.ex_ addition ();
              System.out.println(“ Sum 1 = ” +Sum1);
              System.out.println(“ Sum 2 = ” +Sum2);
             }
}
```