Inheritance

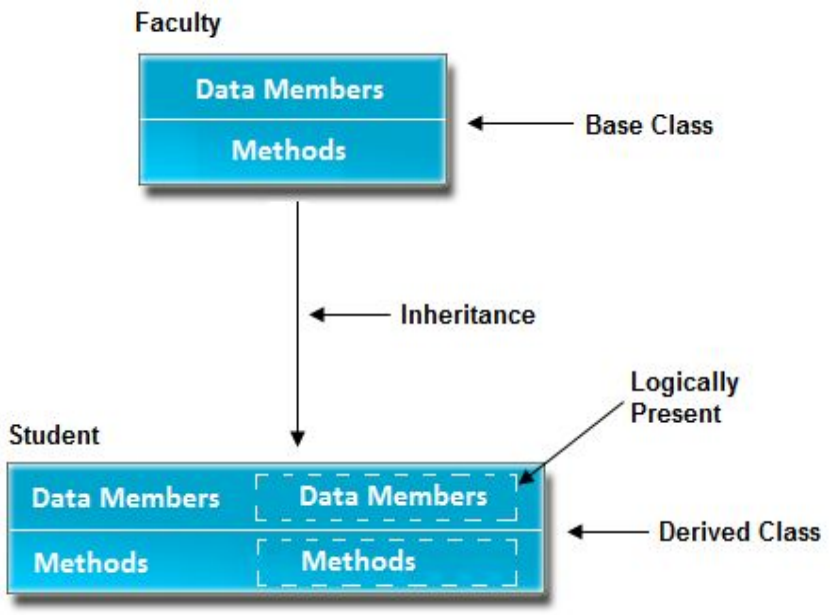
Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object.

The idea behind inheritance in java is that you can create new classes that are built upon existing classes.

When we inherit from an existing class, we can reuse methods and fields of parent class, and we can add new methods and fields also.

extends is the keyword used to inherit the properties of a class.

Inheritance



In the diagram data members and methods are represented in broken line are inherited from faculty class and they are visible in student class logically.

Inheritance

Inheritance represents the **IS-A** relationship, also known as parent-child relationship.

Syntax of Inheritance

```
class Subclass-Name extends Superclass-Name  {
  //methods and fields
}
```

Advantage of Inheritance

If we develop any application using concept of Inheritance then that application have following advantages:

Application development time is less.
Application take less memory.
Application execution time is less.

Advantage of Inheritance

Code reusability: the same methods and variables which are defined in a parent/super/base class can be used in the child/sub/derived class.

Application performance is enhanced (improved).

Redundancy (repetition) of the code is reduced or minimized so that we get consistent results and less storage cost.

Note: In Inheritance the scope of access modifier increasing is allow but decreasing is not allow. Suppose in parent class method access modifier is default then it's present in child class with default or public or protected access modifier but not private(it decreased scope).

Disadvantage of Inheritance

The main disadvantage of using inheritance is that the two classes (parent and child class) get tightly coupled.

This means that if we change the code of parent class, it will affect to all the child classes which are inheriting/deriving the parent class, and hence, it cannot be independent of each other.

Types of Inheritance

Based on number of ways inheriting the feature of **base class into derived class** we have 3 types of inheritance:

Single inheritance
Multilevel inheritance
Hierarchical inheritance

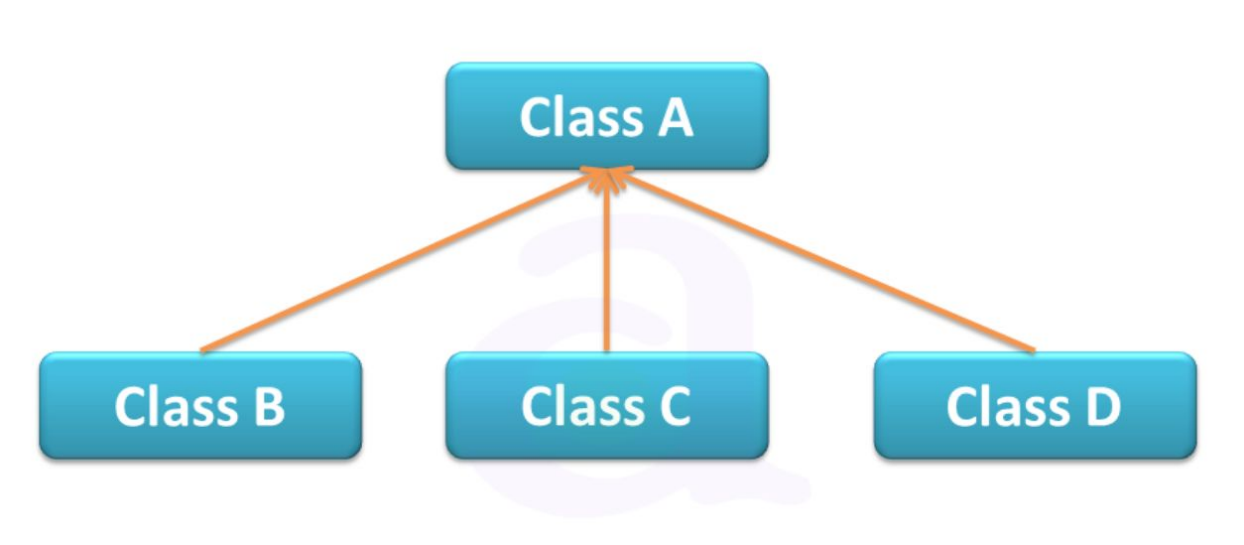
Single inheritance

In single inheritance there exists single base class and single derived class.

```
class Faculty {
   double salary=30000;
}

class Science extends Faculty {
  double bonus=2000;

  public static void main(String args[]) {
     Science obj=new Science();
     SOP("Salary is:"+obj.salary);
     SOP("Bonus is:"+obj.bonus);
  }
}
```

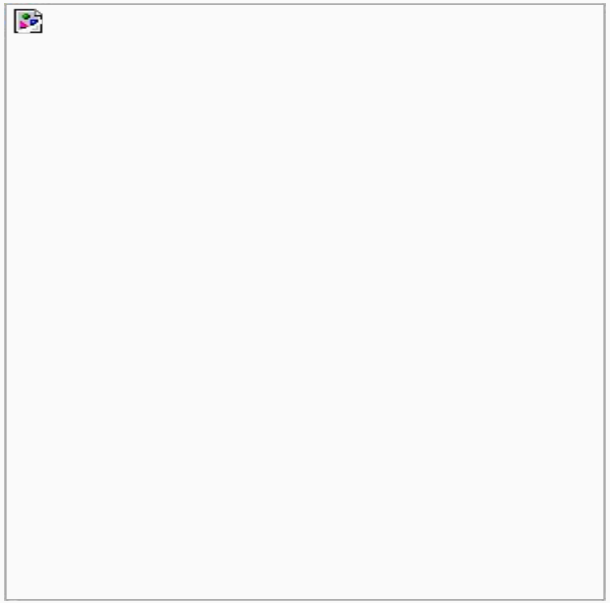
Hierarchical inheritance

When a class has more than one child classes (sub classes) or in other words more than one child classes have the same parent class then this type of inheritance is known as hierarchical inheritance.



Multilevel inheritance

In Multilevel inheritances there exists single base class, single derived class and multiple intermediate base classes.
Single base class + single derived class + multiple intermediate base classes.
Intermediate base classes : An intermediate base class is one in one context with access derived class and in another context same class access base class.

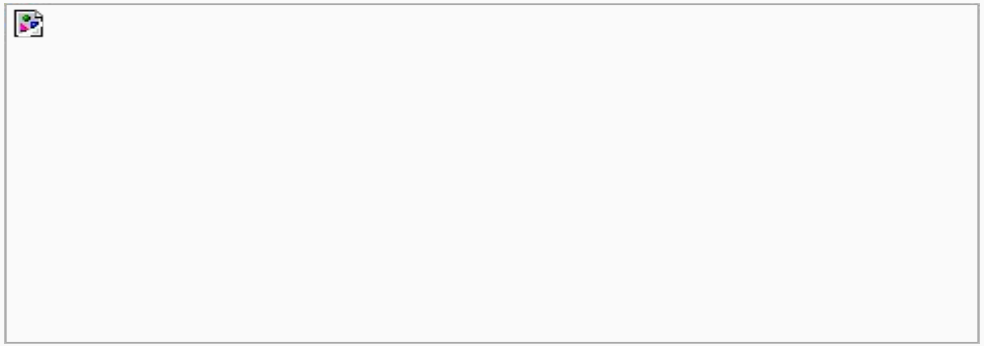


```
class Faculty {
        float total_sal=0, salary=30000;
}

class HRA extends Faculty {
        float hra=3000;
}

class DA extends HRA {
        float da=2000;
}

class Science extends DA {
        float bonus=2000;
        public static void main(String args[]) {
                Science obj=new Science();
                obj.total_sal=obj.salary+obj.hra+obj.da+obj.bonus;
                SOP("Total Salary is:"+obj.total_sal);
        }
}
```

All the above 3 inheritance types are supported by both classes and interfaces

Multiple inheritance

In multiple inheritance there exist multiple classes and single derived class. The concept of multiple inheritance is not supported in java through concept of classes but it can be supported through the concept of interface. To reduce the complexity and simplify the language, multiple inheritance is not supported in java.
Consider a scenario where A, B and C are three classes. The C class inherits A and B classes. If A and B classes have same method and you call it from child class object, there will be ambiguity to call method of A or B class.



Multiple inheritance

Since compile time errors are better than runtime errors, java renders compile time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error now.

```
class A{
    void msg(){SOP("Hello");}
    }
    class B{
    void msg(){SOP("Welcome");}
    }
    class C extends A,B{//suppose if it were

     Public Static void main(String args[]){
       C obj=new C();
       obj.msg();//Now which msg() method
would be invoked?
    }
    }
```

What is not possible in Inheritance?

1. Private members of the superclass are not inherited by the subclass and can only be indirectly accessed.

2. Members that have default accessibility in the superclass are also not inherited by subclasses in other packages, as these members are only accessible by their simple names in subclasses within the same package as the superclass.

3. Since constructors and initializer blocks are not members of a class, they are not inherited by a subclass.

4. A subclass can extend only one superclass