



Inheritance Part :1

Inheritance is a logical process in which one object (Child Class) acquires all the properties and behaviours of a parent object (Parent Class).

Also called as: -

IS-A Relationship

Terms: -

Parent Class → Super Class → Base Class.

Child Class → Sub Class → Derived Class.



extends

1. This Keyword "extends" is used to achieve "inheritance" i.e. to connect the functionality of two different Classes.
2. Properties of Parent Class Are Inherited or given access to the Child Class, which can be accessed through object of Child Class too.
3. Its Specified between Names of Two Classes during the Child Class Declaration but not during Parent Class Declaration.

Syntax:

```
class Parent
```

```
{  
    Parent Class Data Members...  
}
```

```
class Child extends Parent //Child Class Declaration
```

```
{  
    Parent Class Data Members... //Acquired Properties from Parent Class  
    Child Class Data Members...  
}
```

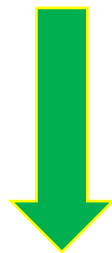


Types of Inheritance

1. Single Level Inheritance.

When One Class Inherits another class is called as Single Level Inheritance.

Class Parent



Class Pandu



Single Level Inheritance:

```
class Parent
{
    void Read()
    {
        System.out.println("Reading");
    }
}
class Pandu extends Parent // Properties of Parent Inherited
{
    void Sing()
    {
        System.out.println("Singing");
    }
}
public class Main
{
    public static void main(String args[])
    {
        Parent pt=new Parent();

        // Object of Parent Class.

        pt.Read();

        // Read() called by Parent Class Object.

        →Error pt.Sing();
```

//Not Allowed, Parent Class object Cannot access Child Class Method.
Properties Flow from top to bottom.
It's a Top Down Approach.



```
Pandu p=new Pandu();  
  
//Object of Child Class.  
  
p.Read();  
  
//Read() Parent Class method called by Child Class Object.  
  
p.Sing();  
  
//Sing() Child Class method called by Child Class Object.  
  
}  
  
}
```

OutPut:

Reading
Reading
Singing



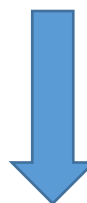
2. Multilevel Inheritance

Multilevel Inheritance in java occurs when a class extends a class that extends another class. This is called multilevel Inheritance in java.

Class Parent



Class Pandu extends Parent



Class Pandi extends Pandu



Smit IT Institute

Gives you wings to fly high



Class Bablu extends Pandi

```
class Parent                                // Level 1
{
    void Read()
    {
        System.out.println("Reading");
    }
}
class Pandu extends Parent // Properties of Parent Inherited    Level 2
{
    void Sing()
    {
        System.out.println("Singing");
    }
}
class Pandi extends Pandu // Properties of Pandu Inherited      Level 3
{
    void Dance()
    {
        System.out.println("Dancing");
    }
}
class Bablu extends Pandi // Properties of Pandi Inherited      Level 4
{
    void Cook()
    {
        System.out.println("Cooking");
    }
}
public class Main                          // Main Class
```



```
{
```

```
public static void main(String args[])
{
    Bablu obj=new Babloo();

    // Only Object of Last Class is Created to access all method at above
    level which are extended.

    obj.Read(); //Properties of Parent Class
    obj.Sing(); //Properties of Pandu Class
    obj.Dance();//Properties of Pandi Class
    obj.Cook(); //Properties of Bablu Class

    *Note: if Base Class Object is Created of any class, it can acquire only
    its own properties but not its child class properties.

}
}
```

OutPut:

Reading
Singing
Dancing
Cooking

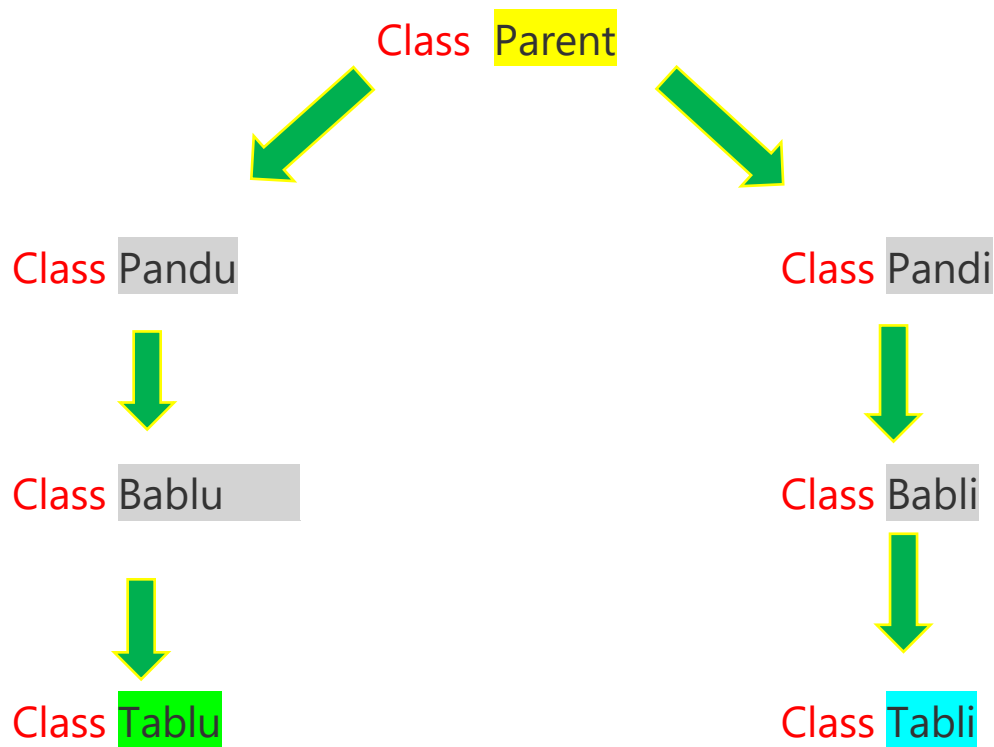


3. Hierarchical Inheritance

1. When two or more classes inherit a single class, it is known as **hierarchical inheritance**.
2. Common Parent Class between Many Child Classes with their own Multilevel Classes can also be present.
3. One Child class Cannot Access the Properties of Another Child Class which is Extending the Same Common Parent Class.



Smit IT Institute
Gives you wings to fly high



Object of **Tablu**

Can access all
Properties from

Tablu→**Bablu**→**Pandu**→**Parent**

Tablu **tu=new** **Tablu()**;

Object of **Tabli**

Can access all
Properties from

Tabli→**Babli**→**Pandi**→**Parent**

Tabli **tu=new** **Tabli()**;



Hierarchical Inheritance

Program: Hierarchical Inheritance.

```
class Parent
{
    void Read()
    {
        System.out.println("Reading");
    }
}
class Pandu extends Parent // Properties of Parent Inherited
{
    void Sing()
    {
        System.out.println("Singing");
    }
}
class Bablu extends Pandu // Properties of Pandu Inherited
SMIT IT CONTACT: 9075127693,
INSTITUTE 7028578967
```



```
{
    void Dance()
    {
        System.out.println("Dancing");
    }
}
class Tablu extends Bablu // Properties of Bablu Inherited
{
    void Cook()
    {
        System.out.println("Cooking");
    }
}

class Pandi extends Parent // Properties of Parent Inherited
{
    void Drawing()
    {
        System.out.println("Drawing");
    }
}
class Babli extends Pandi // Properties of Pandi Inherited
{
    void Riding()
    {
        System.out.println("Riding");
    }
}
class Tabli extends Babli // Properties of Babli Inherited
{
    void Traveling()
```



Smit IT Institute

Gives you wings to fly high

```
{
    System.out.println("Traveling");
}
}
public class Main
{
    public static void main(String args[])
    {
        Tablu tu=new Tablu();

        // Object of Last Class of Parent->Pandu->Bablu->Tablu.
        System.out.println("**Following Properties are \n
        accessed by Object of Tablu Class (Last Class)\n");

        System.out.println("\nProperties of Class Parent.");
        tu.Read();// Properties of Class Parent.

        System.out.println("\nProperties of Class Pandu");
        tu.Sing();//Properties of Class Pandu

        System.out.println("\nProperties of Class Bablu");
        tu.Dance();// Properties of Class Bablu

        System.out.println("\nProperties of Class Tablu");
        tu.Cook();

        System.out.println("\n*****\n");

        Tabli ti=new Tabli();

        // object of Last Class of Parent->Pandi->Babli->Tabli

        System.out.println("**Following Properties are \n
        accessed by Object of Tabli Class (Last Class)\n");

        System.out.println("\nProperties of Class Parent.");
        ti.Read();// Properties of Class Parent.

        System.out.println("\nProperties of Class Pandi");
        ti.Drawing();// Properties of Pandi
```



```
System.out.println("\nProperties of Babli");
ti.Riding();//Properties of Babli.

System.out.println("\nProperties of Tabli");
ti.Traveling(); //Properties of Tabli

}

}
```

OutPut:

**Following Properties are
accessed by Object of Tablu Class (Last Class)

Properties of Class Parent.
Reading

Properties of Class Pandu
Singing

Properties of Class Bablu
Dancing

Properties of Class Tablu
Cooking

**Following Properties are
accessed by Object of Tabli Class (Last Class)

Properties of Class Parent.
Reading

Properties of Pandi
Drawing



Properties of Babli
Riding

Properties of Tabli
Traveling

*Further Notes in Inheritance Part 2