

OBJECT ORIENTED PROGRAMMING USING



Java

Let's explore technology
together to live in the future



Checkout more on
<https://github.com/Sy-hash-collab>



Sy-hash-collab

Inheritance in Java:

Inheritance is a mechanism in Java by which one class is allowed to inherit features (fields and methods) of another class.

Inheritance means creating new classes based on existing ones.

A class that inherits from another class can reuse methods and fields of that class. In addition, you can add new fields and methods to your current class as well.

The new class that is created is known as subclass (child or derived class) and the existing class from where the child class is derived is known as superclass (parent or base class).

The "extends" keyword is used to perform inheritance.

```
class Animal  
{  
    // methods and fields  
}
```

// use of extends keyword to perform inheritance
class Dog extends Animal

```
{  
    // methods and fields of Animal  
    // method and fields of Dog  
}
```

the Dog class is created by inheriting the methods and fields from Animal class.


```
class Vehicle
```

```
{ protected String brand = "Ford"; // Vehicle attribute
```

```
public void honk() { // Vehicle Method  
    System.out.println("Tuut, tuut!");
```

```
} }
```

```
class Car extends Vehicle
```

```
{ // car attribute
```

```
private String modelName = "Mustang";  
public static void main (String [] args)
```

```
{ Car myCar = new Car(); // Creating a myCar Object
```

```
myCar.honk(); // call the honk() method  
                (from Vehicle class) on the  
                myCar object
```

```
System.out.println (myCar.brand);
```

```
// Display value of modelName
```

```
System.out.println (myCar.modelName);
```

```
} }
```

Output:

Tuut, tuut!

Ford

Mustang

• Multi-level inheritance :

A class inherits properties from a class which again has inherits properties.

```
class Shape
{
    public void display()
    {
        System.out.println("Inside display");
    }
}
```

```
class Rectangle extends Shape
{
    public void area()
    {
        System.out.println("Inside area");
    }
}
```

```
class Cube extends Rectangle
{
    public void volume()
    {
        System.out.println("Inside volume");
    }
}
```

```
public class Tester
{
    public static void main (String[] args)
    {
        Cube cube = new Cube();
        cube.display();
        cube.area();
        cube.volume();
    }
}
```


How to create super-class and sub-class?

```
package x;
```

```
public class Human
```

```
{  
    private int eyescount = 2;
```

```
}
```

```
class Teacher extends Human
```

```
{  
    public static void main (String[] args)
```

```
{  
    Teacher t1 = new Teacher();
```

```
    System.out.println (t1.eyescount);
```

```
    }  
}
```

The final keyword: If you don't want others to inherit from a class, use the final keyword.

```
final class Vehicle
```

```
{
```

```
}
```

non-inheritable
can't be parent class
can be a child class

```
class car extends Vehicle // gives error
```

```
{
```

```
}
```