

Java Inheritance

[< Previous](#)[Next >](#)

Java Inheritance (Subclass and Superclass)

In Java, it is possible to inherit attributes and methods from one class to another. We group the "inheritance concept" into two categories:

- **subclass** (child) - the class that inherits from another class
- **superclass** (parent) - the class being inherited from

To inherit from a class, use the **extends** keyword.

In the example below, the **Car** class (subclass) inherits the attributes and methods from the **Vehicle** class (superclass):

Example

```
class Vehicle {
    protected String brand = "Ford";           // Vehicle attribute
    public void honk() {                         // Vehicle method
        System.out.println("Tuut, tuut!");
    }
}

class Car extends Vehicle {
    private String modelName = "Mustang";       // Car attribute
    public static void main(String[] args) {

        // Create a myCar object
        Car myCar = new Car();

        // Call the honk() method (from the Vehicle class) on the myCar
        object
```

```
myCar.honk();

// Display the value of the brand attribute (from the Vehicle class)
// and the value of the modelName from the Car class
System.out.println(myCar.brand + " " + myCar.modelName);
}
```

[Run example »](#)

Did you notice the **protected** modifier in Vehicle?

We set the **brand** attribute in **Vehicle** to a protected access modifier. If it was set to private, the Car class would not be able to access it.

Why Use Inheritance?

- It is useful for code reusability: reuse the attributes and methods of the existing class when you create a new class.

[< Previous](#)[Next >](#)

Copyright 1999-2018 by Refsnes Data. All Rights Reserved.