# #Inheritance and super

As we've recently learned, the `super` keyword is a reference variable that points to the parent class. It's widely used when inheritance comes into the picture, so we're going to need to build some familiarity with it. Of course, our understanding of `super` is dependent upon our understanding of inheritance, so let's get a little practice with that first.

### #Follow the steps below:

1. **Create a `Vehicle` Class (in the right package on Eclipse of course)**
2. Declare String `colour` and String `brand` attributes.
3. Define parameterized constructor for `colour` and `brand` that sets the class attributes.
4. **Create a `Car` class that extends `Vehicle`**
5. Define a parameterized constructor for `colour`, `brand`, and `steeringWheel`
6. Call the `super` keyword appropriately in your constructor (remember, a child class inherits the attributes of its parent class)
7. **Create a `Bike` class that extends the `Vehicle` class**
8. Declare a String `bikeHandle`
9. Define a parameterized constructor for `colour`, `brand`, and `bikeHandle`
10. Call the `super` keyword appropriately in your constructor (remember, a child class inherits the attributes of its parent class)
11. Execute the program and make sure it passes through the test case.

You can see the `Car` class is extending the `Vehicle` class, which means the car is also a vehicle with its own vehicle properties. Inheritance solves a class's problem of reusing properties and helps define the class type. In the above example, a bike and a car are both vehicles.

```
public class Vehicle {
    // Declare instance variables - colour and brand
    String colour;
    String brand;

    // Default constructor
    Vehicle() {
        this("Black", "Tesla");
    }

    // Parameterized constructor
    Vehicle(String colour, String brand) {
        this.colour = colour;
        this.brand = brand;
    }

    public static void main(String[] args) {
        Car car = new Car("White", "Audi", "Chromecoloured");
        Bike bike = new Bike("Grey", "BMW", "Silvercoloured");
```

```
        System.out.println(car.colour + " " + car.brand + " " + car.steeringWh
eel);
        System.out.println(bike.colour + " " + bike.brand + " " + bike.bikeHan
dle);
    }
}

class Bike extends Vehicle {
    // Declare instance variable - bikeHandle
    String bikeHandle;

    // Parameterized constructor for Bike
    Bike(String colour, String brand, String bikeHandle) {
        super(colour, brand);
        this.bikeHandle = bikeHandle;
    }
}
class Car extends Vehicle {
    // Declare instance variable - steeringWheel
    String steeringWheel;

    // Parameterized constructor for Car
    Car(String colour, String brand, String steeringWheel) {
        super(colour, brand);
        this.steeringWheel = steeringWheel;
    }
}
```

# #Using super Keyword with Functions

Follow the steps below:

1. Create a `print` method in your `Vehicle` class: `public void print() { System.out.println(this.colour + " " + this.brand); }`
2. Define a `print()` method in the `Car` class which calls its parent's `print()` method.
3. Create a `print()` method in the `Bike` class which calls its parent's print method.
4. Test your code again using the `Vehicle`'s `main` method.

```
public class Vehicle {

    private String colour;
    private String brand;

    public Vehicle(String colour, String brand) {
        this.colour = colour;
        this.brand = brand;
```

```
    }

    public void print() {
        System.out.println(this.colour + " " + this.brand);
    }

    public static void main(String[] args) {
        Car car = new Car("White", "Audi", "Chromecoloured");
        Bike bike = new Bike("Grey", "BMW", "Silvercoloured");
        car.print();
        bike.print();
    }
}
class Car extends Vehicle {
    String steeringWheel;

    public Car(String colour, String brand, String steeringWheel) {
        super(colour, brand);
        this.steeringWheel = steeringWheel;
    }

    public void print() {
        super.print();
    }
}
class Bike extends Vehicle {
    String bikeHandle;

    public Bike(String colour, String brand, String bikeHandle) {
        super(colour, brand);
        this.bikeHandle = bikeHandle;
    }

    public void print() {
        super.print();
    }
}
```