

## Example 1: Java Inheritance

```
class Animal {

    // field and method of the parent class
    String name;
    public void eat() {
        System.out.println("I can eat");
    }
}

// inherit from Animal
class Dog extends Animal {

    // new method in subclass
    public void display() {
        System.out.println("My name is " + name);
    }
}

class Main {
    public static void main(String[] args) {

        // create an object of the subclass
        Dog labrador = new Dog();

        // access field of superclass
        labrador.name = "Rohu";
        labrador.display();

        // call method of superclass
        // using object of subclass
        labrador.eat();

    }
}
```

## Output

```
My name is Rohu
I can eat
```

## Example 2: Method overriding in Java Inheritance

```
class Animal {

    // method in the superclass
    public void eat() {
        System.out.println("I can eat");
    }
}

// Dog inherits Animal
class Dog extends Animal {

    // overriding the eat() method
    @Override
    public void eat() {
        System.out.println("I eat dog food");
    }

    // new method in subclass
    public void bark() {
        System.out.println("I can bark");
    }
}

class Main {
    public static void main(String[] args) {

        // create an object of the subclass
        Dog labrador = new Dog();

        // call the eat() method
        labrador.eat();
        labrador.bark();
    }
}
```

## Output

```
I eat dog food
I can bark
```

## Example 3: super Keyword in Inheritance

```
class Animal {

    // method in the superclass
    public void eat() {
        System.out.println("I can eat");
    }
}

// Dog inherits Animal
class Dog extends Animal {

    // overriding the eat() method
    @Override
    public void eat() {

        // call method of superclass
        super.eat();
        System.out.println("I eat dog food");
    }

    // new method in subclass
    public void bark() {
        System.out.println("I can bark");
    }
}

class Main {
    public static void main(String[] args) {

        // create an object of the subclass
        Dog labrador = new Dog();

        // call the eat() method
        labrador.eat();
        labrador.bark();
    }
}
```

## Output

I can eat

I eat dog food

I can bark