Trong kế thừa, bạn hoàn toàn có thể ghi đè lại phương thức của lớp cha. Ví dụ:

```
class SuperClass{
	public void display() {
		System.out.println("Hello from SuperClass");
	}
}

class SubClass extends SuperClass{
	@Override
	public void display() {
		System.out.println("Hello from SubClass");
	}
}

class Entry {
	public static void main(String[] args) {
		SubClass s = new SubClass();
		s.display();
	}
}
```

Kết quả khi chạy chương trình:

```
Hello from SubClass
```

Có thể thấy phương thức `display()` của lớp `SuperClass` đã bị ghi đè bởi phương thức `display()` trong lớp `SubClass`. Trong ví dụ trên `@Override` là một annotation để chú thích rằng `display()` là một phương thức ghi đè phương thức từ lớp cha. Việc sử dụng `@Override` annotation là không bắt buộc nhưng trong mọi trường hợp bạn nên dùng nó vì các lợi ích sau:

- Nếu một phương thức được chú thích với annotation `@Override` thì chương trình sẽ kiểm tra xem phương thức này có thực sự ghi đè phương thức của lớp cha không, nếu không thì sẽ báo lỗi. Việc này sẽ giúp bạn ngăn được những lỗi đánh máy như gõ nhầm tên phương thức hoặc nhầm tham số. Nếu không có `@Override` annotation thì bạn sẽ không thể biết được bạn có thực sự đang ghi đè phương thức từ lớp cha hay không.

- Giúp người khác khi nhìn vào code sẽ hiểu được đây là phương thức được ghi đè từ lớp cha.

Lưu ý: Trong trường hợp phương thức của lớp cha bị ghi đè thì bạn vẫn có thể gọi tới nó bằng từ khóa `super`. Ví dụ:

```
class SuperClass{
	public void display() {
		System.out.println("Hello from SuperClass");
	}
}

class SubClass extends SuperClass{
	@Override
	public void display() {
		super.display();
		System.out.println("Hello from SubClass");
	}
}

class Entry {
	public static void main(String[] args) {
		SubClass s = new SubClass();
		s.display();
	}
}
```

Kết quả khi chạy chương trình:

```
Hello from SuperClass
Hello from SubClass
```

Ngoài ra, phương thức `static` sẽ không thể bị ghi đè và đối với các phương thức `non-static` bạn có thể ngăn không cho lớp con ghi đè bằng cách thêm từ khóa `final` giống như sau:

```java
class SuperClass{
    public final void display() {
        System.out.println("Hello from SuperClass");
    }
}

class SubClass extends SuperClass{
    @Override
    public void display() {
        super.di
        System.o
    }
}
```

Cannot override the final method from SuperClass

1 quick fix available:

Remove 'final' modifier of 'SuperClass.display'(..)