**val and var** both are used to declare a variable. **var** is like general variable and its known as a mutable variable in **kotlin** and can be assigned multiple times. **val** is like constant variable and its known as immutable in **kotlin** and can be initialized only single time.

The open annotation on a class is the opposite of Java's final: it allows others to inherit from this class. By default, all classes in Kotlin are final, which corresponds to [Effective Java](http://www.oracle.com/technetwork/java/effectivejava-136174.html), Item 17: *Design and document for inheritance or else prohibit it*.

Overriding Methods

As we mentioned before, we stick to making things explicit in Kotlin. And unlike Java, Kotlin requires explicit annotations for overridable members (we call them *open*) and for overrides:

**open** **class** Base {

**open** **fun** v() {}

**fun** nv() {}

}

**class** Derived() : Base() {

**override** **fun** v() {}

}