Inheritance and Accessibility – revised slide

Any method or data member of a class can be either public, protected, or private.

These modifiers are Java keywords.

This is what they mean:

`public`	inh by:	all subclasses
	acc to:	all classes
`protected`	inh by:	all subclasses
	acc to:	all classes in same pkg
default	inh by:	all subclasses in same pkg
	acc to:	all classes in same pkg
`private`	inh by:	no one
	acc to:	no one

Which level to use? – new slide

If you want your subclasses to inherit a method or data member, you should use `protected`.

Why?

$\boxed{\text{Private:}}$ not inherited.

$\boxed{\text{Public:}}$ inherited by subclasses, but encapsulation is lost because every class has access!

$\boxed{\text{Protected:}}$ inherited, but do get some encapsulation because only acessible to classes in the same package.
Can choose to put only subclasses in the package.

If you haven't used pkgs – revised slide

If you have said nothing about packages in your code, by default every class is considered to be in the same default package, and this is what the modifiers effectively mean:

`public`	inh by:	all subclasses
	acc to:	all classes
`protected`	inh by:	all subclasses
	acc to:	all classes
default	inh by:	all subclasses
	acc to:	all classes
`private`	inh by:	no one
	acc to:	no one

Effectively, your choices are only public or private. With inheritance, you need an in-between level of accessibility for members that only your descendents should have access to.

You need to use packages to get that.