A *class-member-declaration* is permitted to declare a member with the same name or signature as an inherited member. When this occurs, the derived class member is said to hide the base class member. Hiding an inherited member is not considered an error, but it does cause the compiler to issue a warning. To suppress the warning, the declaration of the derived class member can include a new modifier to indicate that the derived member is intended to hide the base member.

Contrary to hiding a name from an outer scope, hiding an accessible name from an inherited scope causes a warning to be reported. In the example

class Base

{

public void F() {}

}

class Derived: Base

{

public void F() {} // Warning, hiding an inherited name

}

the declaration of F in Derived causes a warning to be reported. Hiding an inherited name is specifically not an error, since that would preclude separate evolution of base classes. For example, the above situation might have come about because a later version of Base introduced an F method that wasn't present in an earlier version of the class. Had the above situation been an error, then *any* change made to a base class in a separately versioned class library could potentially cause derived classes to become invalid.

The warning caused by hiding an inherited name can be eliminated through use of the new modifier:

class Base

{

public void F() {}

}

class Derived: Base

{

new public void F() {}

}

The new modifier indicates that the F in Derived is "new", and that it is indeed intended to hide the inherited member.