

Access Modifiers – protected access – how an inherited protected member appears to other classes (i.e. classes using the sub class)



This work by [Rob Cleary](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Based on a work at www.usethetools.net. Permissions beyond the scope of this license may be available via usethetoolsinstructor1@gmail.com

Learning Outcomes:

After doing this exercise, the learner...

1. has been confronted with the specific case of an inherited, protected member, and how this inherited member appears to an unrelated class both inside and outside the same package
2. has had to think deeply about import and package statements and inheritance in Java

Task:

Copy and compile the codes below yourself and experiment briefly to discover what each scenario teaches (oh!, and ask questions).



This work by [Rob Cleary](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Based on a work at www.usethetools.net. Permissions beyond the scope of this license may be available via usethetoolsinstructor1@gmail.com.

Access Modifiers – protected access – how an inherited protected member appears to other classes (i.e. classes using the sub class)

Scenario 1: Protected Access – After inheritance means private (effectively)

Study the code below – and then try to compile it. When you encounter compile errors, reason as to why they occur. Refer to the reading first if possible.

(Reading:

*Sun Certified Programmer for Java 6 Study Guide. 2008. Kathy Sierra and Bert Bates.
pg. 36, paragraph 2 and 3)*

```
package a;
public class A{
    protected void x() {}
}
```

```
package b;
import apackage.A;
public class B extends A
{
    //this class inherits x()
    // but it will appear private to any
    // non-subclass.
    // private void x() {}
}
```

```
package c; //try to change to 'package b;'
import b.B;
class C
{
    public static void main(String[] a)
    {
        B aB = new B();
        aB.x(); //shouldn't compile as x() was protected in A, and so now
               // effectively private in B
    }
}
```

Experiment:

After playing with the above code – try the following: *(try to predict what it does before testing it)*

In class B above add the following method declaration:

```
public void x() {
    x();
}
```

What is it's effect to the overall code?



This work by [Rob Cleary](#) is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Based on a work at www.usethetools.net. Permissions beyond the scope of this license may be available via usethetoolsinstructor1@gmail.com.