Role of Access Modifiers in case of Java Inheritance

Superclass members can be inherited to subclass provided they are eligible by [access modifiers](https://www.scientecheasy.com/2018/12/access-specifiers-in-java-example.html/).

The behaviour of access specifiers in case of inheritance in java is as follows:  
  
1. The private members of the [superclass](https://www.scientecheasy.com/2019/11/java-superclass-subclass.html/) cannot be inherited to the subclass because the private members of superclass are not available to the subclass directly. They are only available in its own class.  
  
2. The default members of the parent class can be inherited to the derived class within the same package.  
  
3. The protected members of a parent class can be inherited to a derived class but the usage of protected members is limited within the package.  
  
4. Public members can be inherited to all subclasses.

Let’s create a program to understand private members of superclass are not accessible in subclass but protected members are available in subclass.

package inheritance;

public class Baseclass

{

private int x = 30;

protected int y = 50;

private void m1()

{

System.out.println("Base class m1 method");

}

protected void m2()

{

System.out.println("Base class m2 method");

}

}

public class DerivedClass extends BaseClass

{

}

public class MainClass

{

public static void main(String[] args)

{

DerivedClass d = new DerivedClass(); // Private members cannot be accessed due to not available in subclass.

d.m2();

System.out.println("y = " +d.y);

}

}

Output:

Base class m2 method

y = 50

Default and public members can be easily accessible in the subclass within the same package.