Documents in this folder:

* MutablePerson.java

Discussion:

I have modified last week’s example to be mutable, and also added a MutablePerson instance variable. Highlight the changes – variables are no longer final, there is now a copy constructor, and every variable now has a setter, making the class no longer immutable.

Demo:

Run MutablePerson.java, and talk through the output. toString and equals make a comeback, but there are also tests to show the behaviour of setters. I have also intentionally added a privacy leak; p3 and p4 both have a reference to p1, rather than a copy, as a result of the constructor. The final tests show that the *parent* objects of both p3 and p4 have been modified, so be sure to highlight the privacy leak.

To fix the leak, change line 93 to make a copy of the given *parent* argument (don’t forget to check for null as well).

Before:  
 this.parent = parent;

After:

If (parent != null) {

this.parent = new MutablePerson(parent);

}

Workshop Notes:

Assessed lab 3 is a little bit different, and a bit trickier than the last two, so make sure to talk about it. Highlight the fact that they will not do well without thinking carefully about how the class should work, and writing appropriate tests for all aspects of that behaviour.

Also explain that marks are awarded for successfully identifying bugs with tests.