

Extra Credit

There is the reason:

In Java, it only allows passing parameters by value.

- 1) Object references are passed by value, which means a copy of the value would be passed to the method as well as the real value of the object.

In the extra credit example of Assignment2, it merely swaps the local object references (employee x, employee y) instead of swapping their properties' values. Therefore, the copied employee x and employee y are swapped. But when the swap method completes execution, the real objects (employee a, employee b) have no change at all.

To make the swap method work, I have written a new swap method in Assignment4.java and test it. It shows that when object references are passed by value, the swap works out!

Expand:

- 2) Immutable object references (example: String, Integer)

If using these classes as object references, the value can't be passed because they are immutable and the field inside is final and can't be changed. However, the local variable can be changed within the method.