

## Questions – Week 2 Lab:

### 1. Why do we use a dedicated Money class instead of storing prices as double?

It allows for editing of prices as time goes by. When using a money class, you can create an object of Money and all the methods attached to that class are connected to the object being created. A dedicated Money class avoids rounding errors, incorrect totals and buggy comparisons by using BigDecimal, which provides exact decimal representation.

### 2. What benefits do we get from enforcing constraints such as “quantity must be greater than zero”?

We get benefits because when the constraints are enforced, the system will output the error handling message disallowing negative values. It stops items being listed at a negative value. This also prevents invalid or nonsensical values (ordering -3 items, charging for -€10)

### 3. Did composition (Order has LineItem) feel more natural than inheritance? Why or why not?

Yes, composition felt more natural because an Order is made up of multiple LineItems. It's a "has-a" relationship, not an "is-a". Inheriting from LineItem would not make sense because an Order is not a special case of a LineItem. Composition allows better modeling of real-world relationships in this case.