**Question 1: Inheritance vs. Composition**

I like to think of inheritance as a classification method. For example, imagine you are unloading the dishwasher and putting away the silverware. You have forks, knives, and spoons, and each type of silverware has a specific slot in the drawer. Here, *silverware* is the parent and *fork*, *knife*, and *spoon* are the children.

The potential drawback with inheritance is that parents and their children are *tightly coupled*. This means that a change in the parent class directly impacts the child, and could break your code. Take the following example: usually when I unload the dishwasher, I put the silverware in the silverware drawer, in the correct slots. However, when it is time to move out of my apartment, I will take the silverware from the dishwasher and put them in a box instead. The box does not have slots, and as such, the child classes will no longer work.

class Silverware {

func toKitchenBox() { }

}

class Fork: Silverware {

func toLeftSlot() { }

}

class Knife: Silverware {

func toMiddleSlot() { }

}

class Spoon: Silverware {

func toRightSlot() { }

}

class Silverware {

func toSilverwareDrawer() { }

}

class Fork: Silverware {

func toLeftSlot() { }

}

class Knife: Silverware {

func toMiddleSlot() { }

}

class Spoon: Silverware {

func toRightSlot() { }

}

*Version 1: Standard*  *Version 2: Moving Day*

Composition, on the other hand, is less about classification and more about building blocks. With composition, each building block provides a particular capability. For example, imagine you are baking a cake (the parent). You will need several types of ingredients (the children): something to provide structure, something to provide sweetness, and something to bind it all together.

Cake!

Binder

*Choose one*

Eggs

Flaxseed + water

Banana

Sweetness

*Choose one*

Granulated Sugar

Powdered Sugar

Brown Sugar

Structure

*Choose one*

All-Purpose Flour

Oat Flour

Almond Flour

**=**

**+**

**+**

As such, composition is a more flexible strategy because parents and their children are *loosely coupled*. In the baking example above, if you are out of one type of flour, you can use another type of flour to serve the same purpose. Similarly, when using composition, changing the components of one object or class in your code will likely not break the rest of your code.

Sources:

<https://betterprogramming.pub/inheritance-vs-composition-2fa0cdd2f939>

<https://www.infoworld.com/article/3409071/java-challenger-7-debugging-java-inheritance.html>