Composite pattern:

Idea:

Consider an army where there are officers, soldiers and a general

What if each officer gives a different order to his regiment will the army be good structured?

So, we must be sure that each officer’s order is the same as the general’s. in other words, we need one source of truth.

The design pattern which allows us to achieve that is the Composite pattern by creating a tree structure where each node can give orders to its children(subordinates).

To force the single source of truth principle we can just let the root(general) give the order and let the order flows through our tree.

Importance and when to use:

* Use it when dealing with a hierarchical business logic like (Army, companies …) and we want to simulate taking orders in that hierarchy
* It is good when it comes to Open-Close principal (Adding and removing classes of objects to the heirarchy)

Relation with other Design patterns:

May be used with:

* Chain of responsibility: pass request to next ones in hierarchy
* Visitor: to loop through the tree and do something

Limitations and pitfalls:

Sometimes, it forces some overgeneralization by treating different classes of objects through the same interface like the (general , officer , soldier) in my example