



## Exercise



# Design Principle Challenge

For each design principle, describe how the Observer Pattern makes use of the principle.

### Design Principle

*Identify the aspects of your application that vary and separate them from what stays the same.*

In the Observer Pattern, the part that changes is the Subject's state and the number and type of Observers.

With this pattern, you can change which Observers depend on the Subject, without changing the Subject itself.

### Design Principle

*Program to an interface, not an implementation.*

Both the Subject and Observer use interfaces. The Subject keeps track of objects implementing the Observer interface, while the observers register with, and get notified by, the Subject interface. As we've seen, this keeps things nice and loosely coupled.

### Design Principle

*Favor composition over inheritance.*

This is a hard one, hint: think about how observers and subjects work together.

The Observer Pattern uses composition to connect Subjects and Observers. A Subject keeps a collection of Observers and notifies them when its state changes. These connections are not fixed by inheritance but are created at runtime through composition.