

Much of software design is about managing dependencies. Avoiding them. Making the ones we need as flexible as possible.

Layers are used to organize dependencies. Facades are used to minimize them.

“Depend on Abstractions, not on Concretions”

Simple Example:

```
// ArrayList is a "Concretion"
ArrayList<Stuff> getStuff();

// List is an "Abstraction"
List<Stuff> getStuff();

// Does order really matter in the result?  If not, use Set
// abstraction
Set<Stuff> getStuff();

// Does uniqueness really matter in the result?  If not, use
// Collection abstraction
Collection<Stuff> getStuff();
```

Similar Example:

```
class StuffManager {

    private ArrayList<Stuff> items;
    // private List<Stuff> items;
    // private Set<Stuff> items;
    // private Collection<Stuff> items;
}
```

Defining Interfaces between Layers

