

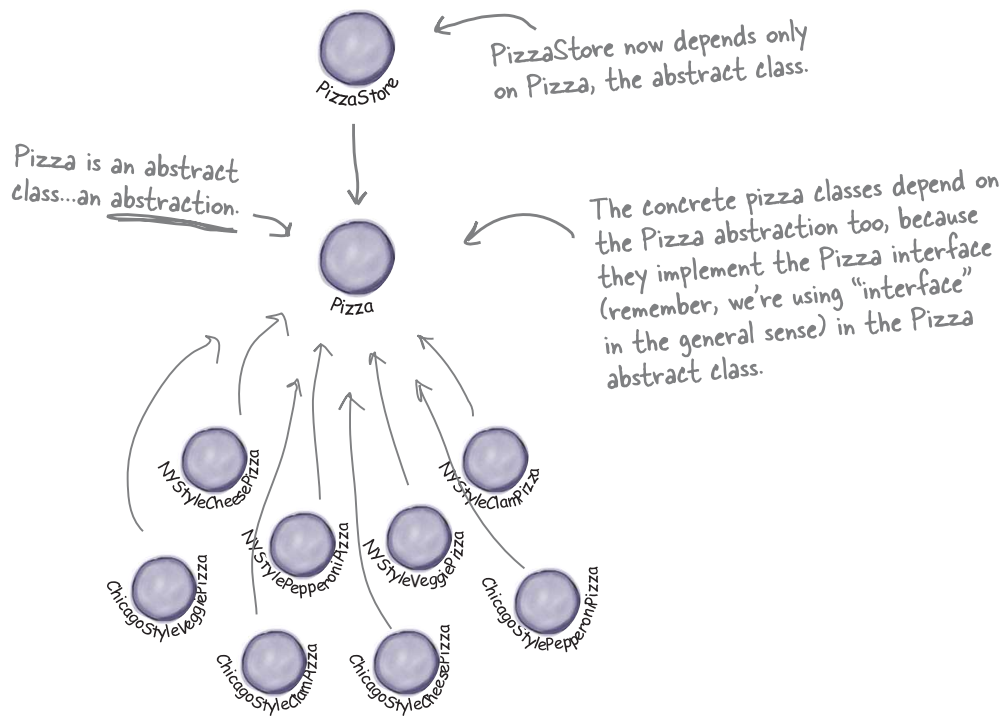
## Applying the Principle

Now, the main problem with the very dependent `PizzaStore` is that it depends on every type of pizza because it actually instantiates concrete types in its `orderPizza()` method.

While we've created an abstraction, `Pizza`, we're nevertheless creating concrete Pizzas in this code, so we don't get a lot of leverage out of this abstraction.

How can we get those instantiations out of the `orderPizza()` method? Well, as we know, the Factory Method Pattern allows us to do just that.

So, after we've applied the Factory Method Pattern, our diagram looks like this:



After applying Factory Method, you'll notice that our high-level component, the `PizzaStore`, and our low-level components, the pizzas, both depend on `Pizza`, the abstraction. Factory Method is not the only technique for adhering to the Dependency Inversion Principle, but it is one of the more powerful ones.