Decorator
Decorator Intent

Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.

(Structural)
Starbucks wants to design an object structure to represent each one of their possible drinks.

Make It Your Drink
2005 Guide to Starbucks® Beverages

How to
If you’re nervous about ordering, don’t be.
There’s no “right” way to order at Starbucks. Just tell us what you want and we’ll get it to you. But if we call your drink back in a way that’s different from what you told us, we’re not correcting you. We’re just translating your order into “barista-speak”—a standard way our baristas call out orders. This language gives the baristas the info they need in the order they need it, so they can make your drink as quickly and efficiently as possible.

“Barista-speak” is easy to learn. It’s all about the order of information. There are five steps to the process.

1. Cup.
2. Shots and size.
3. Syrup.
4. Milk and other modifiers.
5. The drink itself.

Order
1. Cup.
The first thing a barista needs to know is what cup to grab for your drink. If you don’t specify, we’ll put in our 16-oz cup. But you can also ask for a for-here, to-go or personal cup.
2. Shots and size.
Do you want decaf or extra espresso? Here’s something to know: Tall (12 fl oz) drinks usually come with one shot; Grande (16 fl oz) drinks have two. Venti® drinks have two (for 20 fl oz hot drinks) or three (for 24 fl oz cold drinks). So if you add a shot to a Tall, you’re getting a Double Tall.
3. Syrup.
This is the most popular way to customize. We have many different flavors to sweeten or spice up a drink.
4. Milk and other modifiers.
This is when you tell us what milk you want. And if you want something else, like “extra hot” or “extra foamy.”
5. The drink itself.
Don’t forget the most important part! Are you having a latte, a mocha—or something entirely different?

I’d like to have an

ICED, DECAF, TRIPLE, GRANDE, CINNAMON
NO Fat, NO WHIP, MOCHA
MILK AND OTHER MODIFIERS, THE DRINK ITSELF
What is the problem doing this using standard inheritance?

Still to add: Extra Shots
Iced or Hot
Flavoring
A drink starts with a basic drink which then can get “decorated”.

A DrinkElement is a Drink that decorates a Drink.

DrinkElements are Drinks.

These are both Drinks.

Drink::price() + shotPrice
Everything is a Component so the client does not know if it is a ConcreteComponent or a decorated one.

```
Component

operation

Concrete Component

operation()

Concrete Dec #1

addedState

operation()

Concrete Dec #2

addedBehavior()

operation()

Decorator

operation()\n
Component::operation()

addedBehavior()
```

Does this look familiar?