

# Decorator Pattern

Decorator is a structural design pattern in which we try to extend the functionality of the base class object by wrapping it into classes of different functionalities.

For eg:-

Let's say we have a problem of making a system of pizza price calculations with following constraints:-

- There is a base pizza whose price is fixed.
- There are toppings which can be added on base pizza and these costs extra.
- We need to take orders and calculate total, taking above into account.

Base Pizza

Toppings

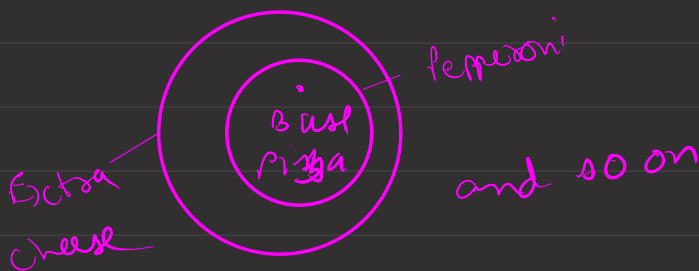
Margherita

Extra cheese

Veggie Delight

Pepperoni

so we can have



### General approach with above example

Eg code :-

```
BasePizza p = new Pepperoni(new ExtraCheese(  
    new Dougherita()))
```

p.cost() → Total cost

### Applicability

- 1 Decorator pattern lets us add more functionalities to basic behaviours at runtime without breaking the code.
- 2 Use it when you are not able to extend an object's behaviour using inheritance