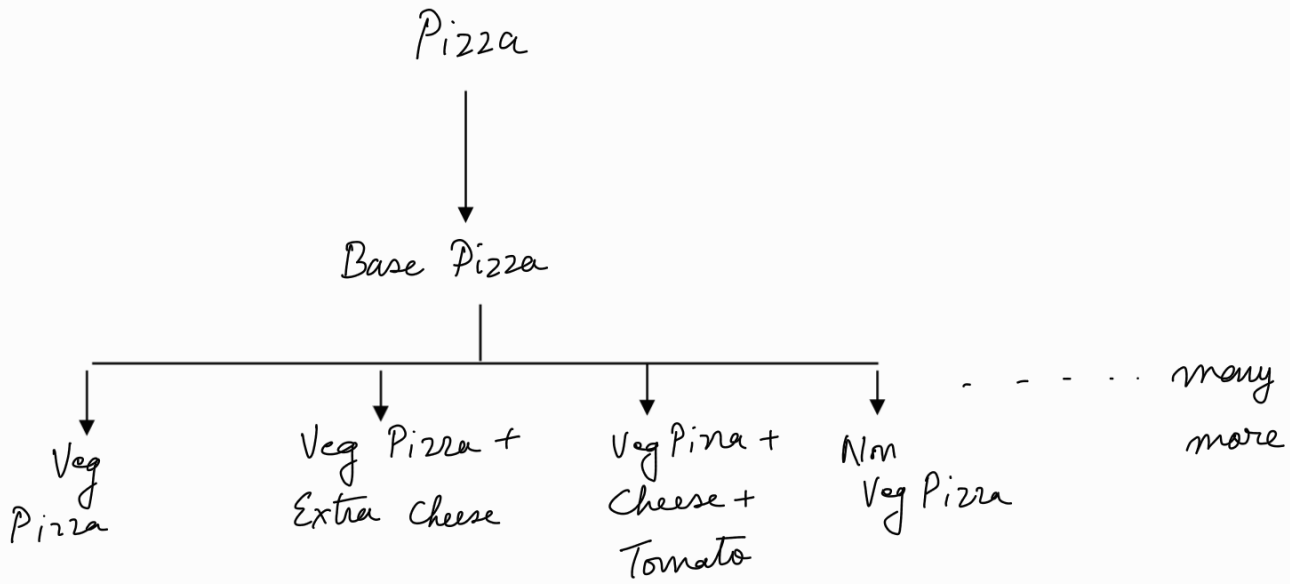


```
interface Pizza {
        bake();
}
```

```
BasePizza impl. Pizza {
        bake();
}
```

```
PizzaDecorator impl. Pizza {
        Pizza pizza;
        constructor(Pizza);
        bake();
}
```

```
ExtraCheese extends PizzaDecorator {
        Pizza;
}
```

* it is a structural pattern that lets you add new behaviour to objects dynamically without altering their structure. It is useful for adhering to open/closed principle.

- Component : the interface defining the base behaviour
- Concrete Components : the original object to which additional behaviour can be added.
- Decorator : abstract class that implements the component interface and contains a reference to a component.
- Concrete Decorator : adds new behaviour to a base component.