Use the ***Chain of Responsibility Pattern*** when you want to give more than one object a chance to handle a request.

***Scenario – Mail service for Gumball Machine:***

With the Chain of Responsibility Pattern, you create a chain of objects to examine requests. Each object in turn examines a request and either handles it, or passes it on the next object in the chain.

NewLocHandler

handleRequest()

ComplaintHandler

handleRequest()

FanHandler

handleRequest()

SpamHandler

handleRequest()

Handler

Successor

handleRequest()

Each email is passed into first handler

***Benefits:***

* Decouples the sender of the request and its receivers.
* Simplifies your object because it doesn’t have to know the chain’s structure and keep direct references to its members.
* Allows you to add or remove responsibilities dynamically by changing the members or order of the chain.

***Uses and Drawbacks:***

* Commonly used in windows systems to handle events like mouse clicks and keyboard events.
* Execution of the request isn’t guaranteed; it may fall off the end of the chain if no object handles it (this can be an advantage or a disadvantage).
* Can be hard to observe or debug at runtime.