**Name:** Chain of Responsibility

**Category:** Behavioral

**Description:** Chain of Responsibility is a software engineering design pattern that is used to pass requests along a chain of objects or handlers. The client sends a request to an object, and the object will either handle the request or pass it along to the next object in the chain. The request is passed down the chain until an object handles it. The initial request does not have to be sent to the first object in the chain, and the “handler” of the request is determined at runtime. It is also possible for a request to not go through the chain at all and for a handler to stop the request before it is fulfilled.

A common real-world example of Chain of Responsibility is contacting any form of online support and being passed down a “chain” of support agents or employees until one finally gives you the help you need. When you call your local electronics store to *request* that they check their stock for a specific video game, you are connected with a robot that asks you to press a digit based on which department you wish to be referred to. The robot cannot *handle* your *request* itself, so it *passes* your *request* to an actual person in the store who then *handles* that *request*. It is also possible that your *request* gets *passed* amongst several employees or *handlers* until one is able to properly *handle* your *request*.

Another example of Chain of Responsibility is the process of going through an airport to be cleared for your flight. You, the client, have the request of boarding the airplane. First your request must be passed to the check-in counter. They verify your identity and print your boarding pass but cannot allow you on the plane just yet, so they pass your request onto the second handler, TSA. TSA will check you and your carry-on items but cannot handle your request either, so they will pass you onto immigration and passport control. This handler will check your visa, give you a stamp on your passport, but cannot let you on the plane. Next your request is passed to customs inspection. Here, they will ensure you have nothing that is not allowed in your destination country but can still not put you on the plane. Finally, your request is passed to the boarding gate which then fulfills your request and lets you aboard the plane. In any of these steps, the handler could have decided that you did not pass their checks and stopped the chain without you getting your request fulfilled.

**When to Use:** Chain of Responsibility is often used in cases where you want to keep the sender and receiver of the request separated, you have multiple objects that need to be processed the same way or could all qualify to fulfill the request, you want to process requests through a series of authorization checks, or you want to make it easier to modify handlers without rewriting the whole program.

**Advantages:** flexible, reusable, dynamic

**Disadvantages:** hard to debug, has potential to become unnecessarily long and complex