The Bridge Pattern is part of the Structural Design patterns. As the name may suggest, it acts as an intermediary between two components. But if we talk about the Adapter Pattern then both patterns have the same logical definition. To say that it acts as an intermediary is partially correct. Both are a type of intermediary between two systems, but the actual difference is the goal that these patterns achieve. The Adapter Pattern is designed to act as an intermediary when the two components are not compatible to work with each other. Also the Adapter Pattern can add more functionality to a source component request, before passing it on to the target component, with which the source component is to interact. On the other hand, the Bridge Pattern's purpose is to provide multiple pathways between two components that are nothing but achieving many to many communication between multiple implementations of the source and the target components. So technically, it receives a request from one of the many implementations of the source component, and based on the client requirements, sends it to one of the many implementations of the target component.

As per the preceding definition, it helps us to create a structure, where even the interface is separated from the implementation using a bridge. Technically this results in a system where a function can have multiple implementations and each implementation of that function can be used in multiple ways. So this results in a system with many-to-many mappings. Let's explain this using a real-world example.