Bridge Design Pattern

A Bridge Pattern says that just **"decouple the functional abstraction from the implementation so that the two can vary independently".**

The Bridge design pattern allows you to separate the abstraction from the implementation. It is a structural design pattern.

**There are 2 parts in Bridge design pattern :**

1. Abstraction
2. Implementation

The Bridge Pattern is also known as **Handle or Body.**

Advantage of Bridge Pattern

* It enables the separation of implementation from the interface.
* It improves the extensibility.
* It allows the hiding of implementation details from the client.

Usage of Bridge Pattern

* When you don't want a permanent binding between the functional abstraction and its implementation.
* When both the functional abstraction and its implementation need to extended using sub-classes.
* It is mostly used in those places where changes are made in the implementation does not affect the clients.

Example of Bridge Pattern

The UML given below describes the example of bridge pattern.

UML for Bridge Pattern:

