**Design Pattern:-**

**Scenario:**

Let us suppose there is Manufacturing Company which manufactures two types of Cars namely Suzuki and Honda. If I decide to buy any of the cars, I need to know details of each car based on their name. If I select Suzuki, I should be able to get all the information about it and similarly for Honda. This can be achieved by directly creating two simple classes but if suppose company decides to launch a new Car, we need to modify the code and add new Car as well as the logic to get the details of that Car. Let us say 100 cars are added in the list, so we need to write the logic for 100 cars in the code.

**Resolution:**

**Factory Design Pattern** is a part of creational pattern. Using this pattern we can create instance of the class without exposing the logic to the outer world. Factory pattern allows us to define the class of the object at runtime. It also allows to encapsulate object creation so that we can keep all object creation code in one place.

For solution of the above problem, if we use factory design pattern there is no worry about the logic. We just need to write a new class for new Car and add logic to create just the instance of the class.