**Factory Design Pattern (FDP):**

**Link:** https://www.youtube.com/watch?v=blDsmrsyOME&list=PLk6CEY9XxSIDZhQURp6d8Sgp-A0yKKDKV&index=2)

**DESCRIPTION:**

Factory Design Pattern is very basic design pattern and it is one of the creational design patterns.

In this design pattern we create a factory method and we pass type to this method and it returns that type's object.

1. It creates object for you, rather you initiate object directly. (It creates object at Runtime)

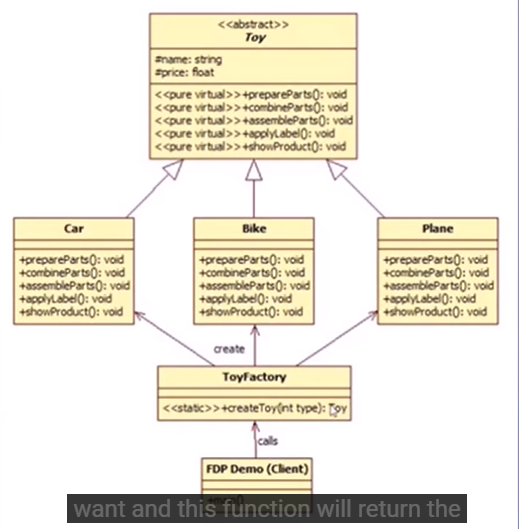
2. FDP is also known as "virtual constructor"

{ c++ doesn't support virtual constructor }

**How to Implement FDP:**

Define an interface or an abstract class for creating an object but

let the subclasses decide which class to initiate



**Advantages:**

Less code change if we change object creation process

we create object without exposing creation logic to the client

we get the benefits of virtual constructor // we get object creation at runtime

**More about Factory Pattern:**  
https://refactoring.guru/design-patterns/factory-method