**Factory Design Pattern**

**Factory Pattern**

**Define Factory Design Pattern:**

A Factory Pattern or Factory Method Pattern says that just define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate. In other words, subclasses are responsible to create the instance of the class.

The Factory Method Pattern is also known as Virtual Constructor.

**Real-Life Example of Factory Pattern:**

As we know Factory is used to manufacture something as per requirement and if new item is going to add to manufacture with Factory. Factory will also manufacture those items as well. Factory class provides abstraction between client and Car when create the instance of the Car [Honda, BMW].

**When to use the Factory Design Pattern in real-time applications:**

To reduce tight coupling between client and the product.

This design pattern provides the client with a simple mechanism to create the object. So, we need to use the Factory Design Pattern in C# when:

* The Object needs to be extended to the subclasses
* Classes don’t know what exact sub-classes it has to create
* The Product implementation going to change over time and the Client remains unchanged.

##### ****Problems of Simple Factory Pattern in C#:****

##### **If we need to add any new product (i.e. new credit card) then we need to add a new if else condition in the GetCreditCard method of the CreditCardFactory class. This violates the open/closed design principle.**

##### **We also have a tight coupling between the Factory (CreditCardFactory) class and product classes (MoneyBack, Titanium, and Platinum).**

**Factory Method Design Pattern**

**Define Factory Method Design Pattern:**

Define an interface for creating an object, but let the subclasses decide which class to instantiate. The Factory method lets a class defer instantiation it uses to subclasses. Without exposing the object creation logic to the client.

To achieve this, in the factory method design pattern we will create an abstract class as the Factory class which will create and return the instance of the product, but it will let the subclasses decide which class to instantiate.

**Abstract Factory Design Pattern**

**Define Factory Method Design Pattern:**

The Abstract Factory Design Pattern provides a way to encapsulate a group of individual factories that have a common theme without specifying their concrete classes.

In simple words we can say, the Abstract Factory is a super factory that creates other factories. This Factory is also called Factory of Factories.

**Pointe to Remember:**

* Abstract Factory Pattern provides an interface for creating families of related dependent objects without specifying their concrete classes.
* The Abstract Factory Pattern provides a way to encapsulate a group of individual factories that have a common theme without specifying their concrete classes.
* The abstract factory design pattern is merely an extension to the factory method pattern or factory pattern, which allows you to create objects without being concerned about the actual class of the object being created.
* Abstract means hiding some information and factory means which produces the products and pattern means a design. So, the Abstract Factory Pattern is a software design pattern that provides a way to encapsulate a group of individual factories that have a common theme.

**When to use it Abstract Factory Design Pattern:**

* When you want to create a set of related objects or dependent objects which must be used together.
* When the system should configure to work with multiple families of products.
* When the Concrete classes should be decoupled from the clients.

**Differences between Abstract Factory and Factory Method Design Pattern:**

* Abstract Factory Design Pattern adds a layer of abstraction to the Factory Method Design Pattern
* The Abstract Factory design pattern implementation can have multiple factory methods
* Similar products of a factory implementation are grouped in Abstract factory
* The Abstract Factory Pattern uses object composition to decouple applications from specific implementations
* The Factory Method Pattern uses inheritance to decouple ( microservice ) applications from specific implementations