**Abstract Factory**

Creates an instance of several families of classes.

Provide an interface for creating families of related or dependent objects without specifying their concrete classes.

Abstract factory pattern in useful when the client needs to create objects, which are somehow related. If we need to create the family of related or dependent objects, then we can use Abstract Factory Pattern.

This pattern is particularly useful when the client does not know exactly what type to create. As an example, let's say a Showroom exclusively selling cellphones gets a query for the smart phones made by Samsung. Here we don't know the exact type of object to be created (assuming all the information for a phone is wrapped in the form of a concrete object). But we do know that we are looking for smart phones that are manufactured by Samsung. This information can actually be utilized if our design has Abstract factory implementation.

So with this idea of Abstract Factory pattern, we will now try to create a design that will facilitate the creation of related objects. We will go ahead and write a rudimentary application for the scenario we just talked about.

An interface for creating families of related or dependent objects without specifying their concrete classes. We can say it is just an object maker, which can create more than one type of object.

The object it produces is known to the client only by that object's interface, not by the object's actual concrete implementation.

We use it when we have a requirement to create a set of related objects, or dependent objects which must be used together as families of objects. Concrete classes should be decoupled from clients.

UML class diagram

