**Factory design pattern:**

Factory design pattern is used to create objects or [Class in Java](http://javarevisited.blogspot.com/2011/10/class-in-java-programming-general.html) and it provides loose coupling and high cohesion. Factory pattern encapsulate object creation logic which makes it easy to change it later when you change how object gets created or you can even introduce new object with just change in one class. In GOF pattern list Factory pattern is listed as Creation design pattern. Factory should be an interface and clients first either creates factory or get factory which later used to create objects.

Let’s see an example of how factory pattern is implemented in Code.We have requirement to create multiple currency e.g. INR, SGD, USD and code should be extensible to accommodate new Currency as well. Here we have made Currency as interface and all currency would be concrete implementation of Currency interface. Factory Class will create Currency based upon country and return concrete implementation which will be stored in interface type. This makes code dynamic and extensible.

Why we use this :

1. Encapsulation : don’t want to complexity of objection creation for security purpose.
2. Complexity hiding.
3. Open for extension and closed for modification.
4. Object creation will be simple for client (like based on mobile type we will provide Mobile Object)

**examples of Factory Method:**

1. Say, in a ‘Drawing’ system, depending on user’s input, different pictures like square, rectangle, circle can be drawn. Here we can use factory method to create instances depending on user’s input. For adding new type of shape, no need to change client’s code.
2. Another example: In travel site, we can book train ticket as well bus tickets and flight ticket. In this case user can give his travel type as ‘bus’, ‘train’ or ‘flight’.  
   Here we have an abstract class ‘AnyTravel’ with a static member function ‘GetObject’ which depending on user’s travel type, will create & return object of ‘BusTravel’ or ‘ TrainTravel’. ‘BusTravel’ or ‘ TrainTravel’ have common functions like passenger name, Origin, destinationparameters.

<https://javarevisited.blogspot.com/2011/12/factory-design-pattern-java-example.html>

<https://javarevisited.blogspot.com/2012/03/what-is-encapsulation-in-java-and-oops.html>

<https://www.geeksforgeeks.org/design-patterns-set-2-factory-method/>

<https://www.youtube.com/watch?v=KgfpICxwOOM>