

Factory Method:

- Factory Method is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created. In Factory pattern, we create objects without exposing the creation logic to the client and the client uses the same common interface to create a new type of object.
- The idea is to use a static member-function (static factory method) that creates & returns instances, hiding the details of class modules from the user.
- A factory pattern is one of the core design principles to create an object, allowing clients to create objects of a library in a way such that it doesn't have tight coupling with the class hierarchy of the library.
- The library is responsible to decide which object type to create based on an input. Client just needs to make calls to the library's factory Create method and pass the type it wants without worrying about the actual implementation of the creation of objects.

Examples of Factory Method:

Say, in a 'Drawing' system, depending on the user's input, different pictures like square, rectangle, the circle can be drawn. Here we can use the factory method to create instances depending on the user's input. For adding a new type of shape, no need to change the client's code.