* I am using Factory design pattern. By doing this we can get the subclasses to provide extended version of an object, since creating an object inside a subclass is more flexible than creating individually of each class. Also, it encourages consistency of code since we now create object using factory rather than using constructors of each class. Moreover, since all the objects are created on a centralized platform(factory) it is easier to debug and maintain the code. It hence results in loose coupling of the code and reduces dependencies. We thus don’t hinder the already running/deployed code and just add new class/factory. In Singleton design pattern we make sure that the class can have only one instance and it can be accessed globally. It saves memory because object is created once. Only that instance is reused everytime.