Exercise:

Let’s take about a real-world problem and the solution at the same time with the help of Prototype Design Pattern.

Let’s say we must open an apparel store in which clothes are the main product. Now, the owner decides to open the same outlet of the store at a different location.

This Store setup and opening can be done in two ways, one is to recreate the whole store from scratch and the other is just clone the previous store which is already a fully established store.

From the above two approaches, the second one is the easiest one and cost-effective as well. In this way, the store can be opened in a very small amount of time and effort.

Hint of a way to solve the problem in our code:  
  
We need a class Apparelstore.java which is the Store class that is having two variables storename and the list of the clothes which are present in the store. To fill the store’s product which is clothes in our case, with the data we are using the fetchdata() method, this method is setting the value of clothes with some custom logic.

As mentioned in the above we can use any type of object cloning technique so, I recommend to use a **Deep copy** because it is the suggested technique for the prototype design pattern. The reason behind this is that our cloned object should be independent with the changes in the prototype object(Original Object).

While doing the Deep copy, we need to override the clone() method and create a new apparelstore object. After this, in the same object, we need to copy the values of Clothes object into the newly created list of clothes. At last, the cloned object will be returned by the clone() method.

The PrototypeMain.java is our main class in which we are trying to copy the object from the prototype object. At first, we have created a new store and set the name of the store. fetchData()will set the data into the clothes list the prototype store will be created.

Again, we try to create the other store which will be a copy of the prototype store. We need to call the clone() which we have created in the Apparelstore.java class. This will create a completely new outlet of the store with the same clothes that were available in the prototype store.

Here you can check the solution in details:

[Prototype Design Pattern in Java - Complete Tutorial | Jstobigdata](https://jstobigdata.com/java/prototype-design-pattern-in-java-with-a-real-world-example/)