What is ORM?
  It stands for Object Relational Mapping.
  The process of linking DB tables with java classes(BO) and DB table column names with properties of classes and
  having synchronization b/w them is called "OR-Mapping".

Synchronization
  The modification done in the Object of java classes will reflect to DB table records and vice versa.

      ORM tools ===> hibernate,ibatis,eclipselink,.....

SpringORM
=========
 => It is not another ORM-Framework
 => It is a SpringModule which provides abstraction on multiple orm frameworks like hibernate,ibatis etc
 => It is available to simplify objects based on O-R mapping persistence logic.
 => It provides multiple template classes like "HibernateTemplate" which is given to avoid boiler plate code
    of O-R mapping persistence logic.

Plain hibernate
===============
1. Create a Configuration Object(To activate hibernate s/w)
2. Create a Session Factory Object
3. Create a Session Object
4. Begin Transaction

5. Perform persistence operation [ApplicationLogic]

6. use commit/rollback operation
7. close session/sessionfactory objects

In the above steps 1,2,3,4,6 and 7 is a common operation which is referred as "Boiler plate code".
Using SpringORM all the "Boiler plate code" steps will be taken care by a Template class called "HibernateTemplate".

Spring ORM Code(Integration with hibernate)
===
1. Create an Inject Hibernate Template class object
2. Perform Persistence operation
                  Note: boiler plate problem is avoided.

Advantages of Spring ORM
========================
1. Avoids boiler plate code by suppling Template classes
2. Common Exception handling(we need to just handler ORM specification excpetion==> DataAccessException)
3. Persistence logic is portable across multiple DB S/w and Entity classes are also portable across multiple ORM Frameworks.
4. Common Transaction Management Support
5. Common Single Row Operation(given by JPA) and Common JPQL(Java Persistence Query Language)