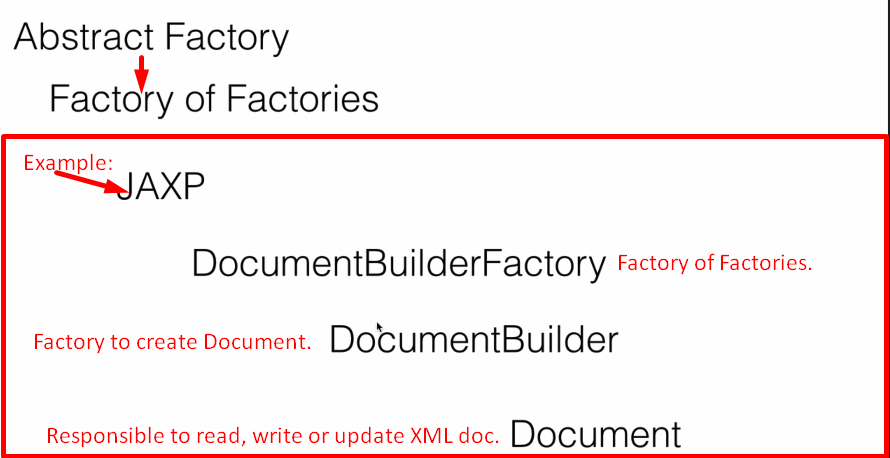
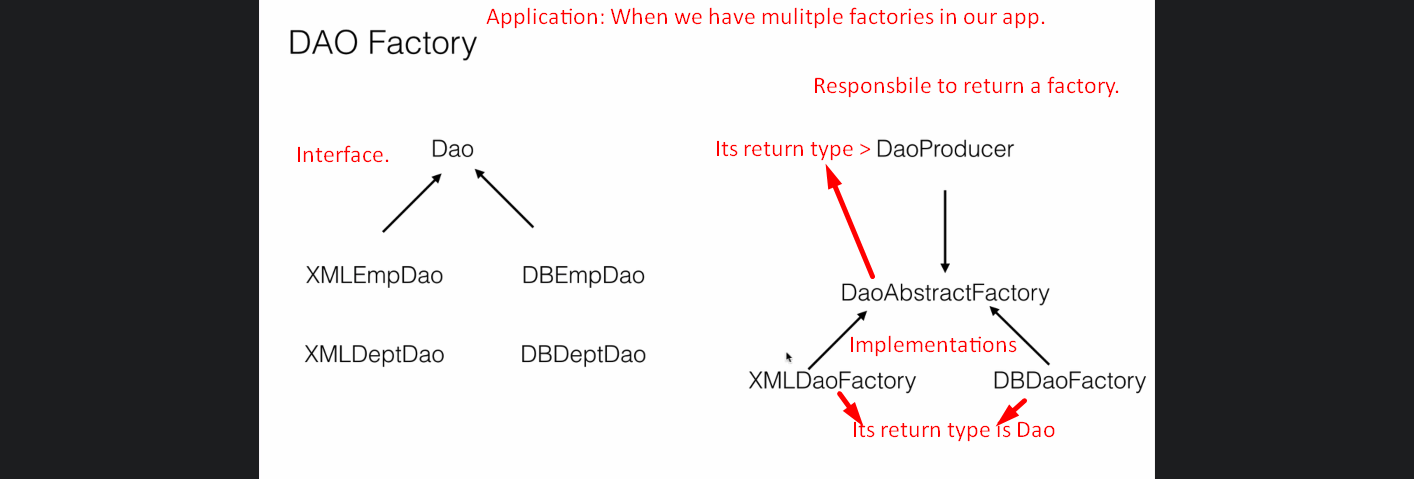
1. **Abstract Factory Pattern**: Hides the creation of the factory.
2. **Example**: Good example in Java Space is JAXP.
   1. **JAXP**: Java API for XML Parsing.
   2. Using this API, we can read, write or update XML from an xml file.
   3. Key class is **Document** representing XML document in memory.
   4. To create a **Document** class, we use **DocumentBuilder** which is a factory class.
   5. There is one more class **DocumentBuilderFactory** responsible for creating **DocumentBuilder**.
   6. **DocumentBuilderFactory** is abstract factory because this is factory of factories.
   7. 
3. The Use Case, we’re going to work on is DAO Factory.
   1. Dao responsible for CRUD operations.
   2. We will have different kinds of DAO.
      1. XML Dao.
      2. DB Dao
   3. Within XML, we can have Employee or Department Data so XMLEmpDao, XMLDepDao.
   4. Within DB, we can have Employee or Department Data so DBEmpDao, DBDepDao.
   5. We will have two factories which would give a specific Dao.
      1. XMLDaoFactory.
      2. DBDaoFactory.
   6. DaoAbstractFactory would return either XMLDaoFactory or DBDaoFactory.
4. 
5. **NOTE**: See the next lecture for UML diagram if above slide is not clear.