* Used to create familes of related objects without specifying their concrete classes.
* It provides an interface for creating objects ,but actual instantiation is delegated to concrete subclasses known as factories.
* Used when system needs to be independent on how its objects are created or when it needs to support multiple familes of objects .
* Abstract Factory provides flexible weay to create families of objects and makes it easier to introduce new product families without changing linet code

Abstract Factory — this is a factory design pattern with extra level of abstraction. Examples are:

* [javax.xml.parsers.DocumentBuilderFactory](https://docs.oracle.com/en/java/javase/15/docs/api/java.xml/javax/xml/parsers/DocumentBuilderFactory.html) — newInstance()
* [javax.xml.transform.TransformerFactory](https://docs.oracle.com/en/java/javase/15/docs/api/java.xml/javax/xml/transform/TransformerFactory.html)— newInstance()
* [javax.xml.xpath.XPathFactory](https://docs.oracle.com/en/java/javase/15/docs/api/java.xml/javax/xml/xpath/XPathFactory.html) — newInstance()