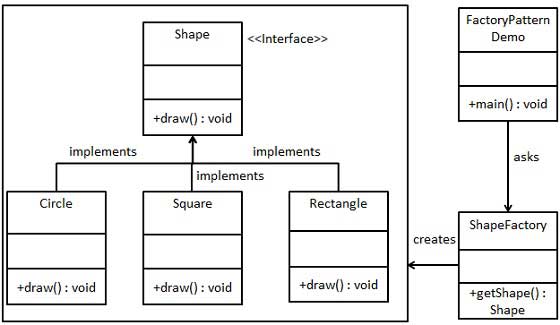
**Difference between AbstractFactory and Factory design patters are as follows:**

* **Factory Method** is used to create one product only but **Abstract Factory** is about creating families of related or dependent products.
* **Factory Method** pattern exposes a method to the client for creating the object whereas in case of **Abstract Factory** they expose a family of related objects which may consist of these Factory methods.
* **Factory Method** pattern hides the construction of single object where as **Abstract factory method** hides the construction of a family of related objects. Abstract factories are usually implemented using (a set of) factory methods.
* **AbstractFactory** pattern uses composition to delegate responsibility of creating object to another class while **Factory** design pattern uses inheritance and relies on derived class or sub class to create object.
* The idea behind the **Factory Method** pattern is that it allows for the case where a client doesn't know what concrete classes it will be required to create at runtime, but just wants to get a class that will do the job while **AbstractFactory** pattern is best utilised when your system has to create multiple families of products or you want to provide a library of products without exposing the implementation details.!

**Factory Method Pattern Implementation:** 

**AbstractFactory Pattern Implementation:**

