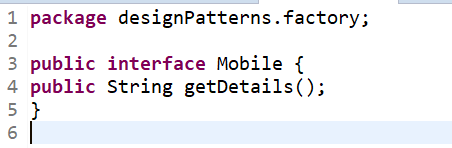
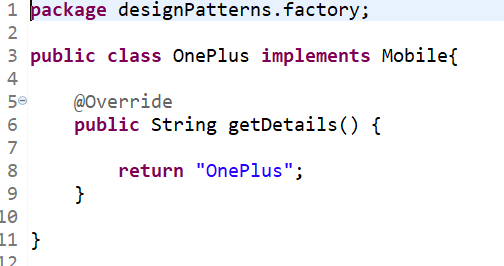
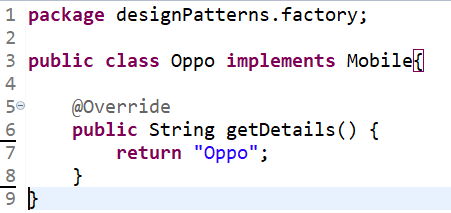
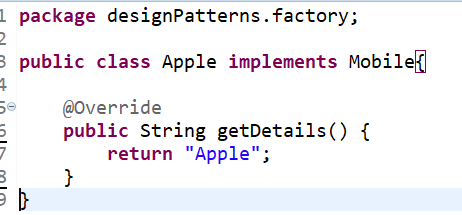
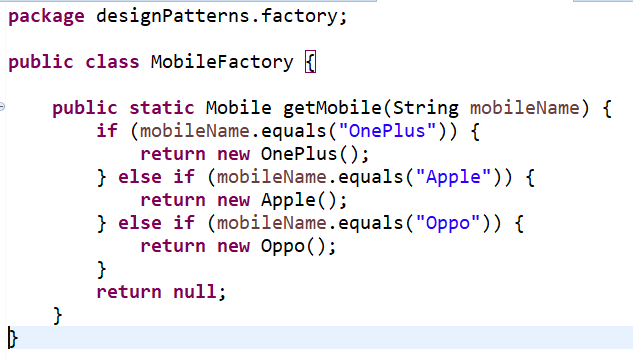
The factory design pattern is used to define a method that creates objects, but lets subclasses decide which class to instantiate. The factory method design pattern lets a class defer instantiation to subclasses.

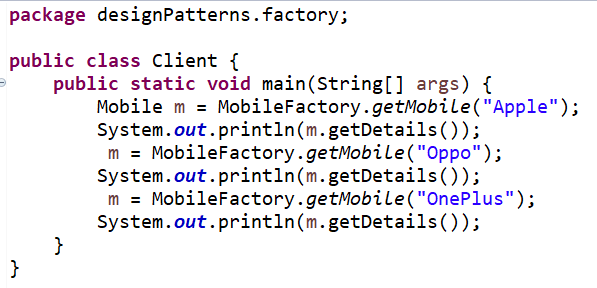












**Difference between Dependency Injection and Factory Pattern.**

Factory and Dependency injection both are the design pattern which can be used to enhance loose coupling abilities between the software components.

Factory design pattern is used to create objects. But, injection and life cycle management of the object should be handled by programmer within the application. There is no way to configure everything in a single place. Therefore, programmers need to call object creation logic wherever it needed which eventually hinder the loose coupling abilities.

In DI design pattern, creation of object, injecting of the instance and life cycle management of the instance can be handled outside the code. In spring, Instantiation ,Injection and Lifecycle management are handled by IOC container.

| **Sr. No.** | **Key** | **Factory design pattern** | **DI design pattern** |
| --- | --- | --- | --- |
| 1 | Object Creation | It is used to create objects. We have separate Factory class which contains creation logic. | It is responsible for creation and injection of the object. |
| 2 | State Of the Object | It is responsible for creation of stateful objects. | It is responsible to create stateless objects |
| 3 | Runtime /Compile time | Create object at compile time | Configure objects at runtime |
| 4 | Code Change | In case of change in business requirements, object creation logic may be changed. | No code change required |
| 5 | M echanism | Class is dependent on factory method which in turn have dependency on concrete classes | Parent object and all dependent object can be created at single location |