***Factory Method Pattern:***

The Factory Method Pattern defines and interface for creating an object, but lets subclasses decide which class to instantiate. Factory method lets a class defer instantiation to subclasses.

***Abstract Factory Pattern:***

The Abstract Factory Pattern provides an interface for creating families of related or dependent objects without specifying their concrete classes.

* All factories encapsulate object creation
* Simple Factory, while not a bonafide design pattern, is a simple way to decouple your clients from concrete classes.
* Factory method relies on inheritance: object creation is delogated to subclasses, which implement the factory method to create objects.
* Abstract Factory relies on object composition: object creation is implemented in methods exposed in the factory interface
* All factory patterns promote loose coupling by reducing the dependency of your application on concrete classes.
* The intent of Factory Method is to allow a class to defer instantiation to its subclasses.
* The intent of Abstract Factory is to create families of related objects without having to depend on their concrete classes.
* The Dependency Inversion Principle guides us to avoid dependencies on concrete types and to strive for abstractions.
* Factories are a powerful technique for coding to abstractions, not concrete classes.