Factory Method

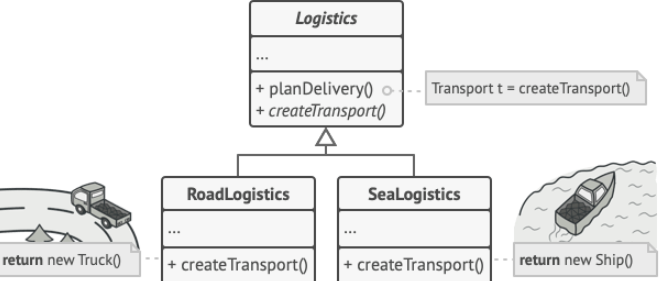
Creational Design Pattern

**What problems does Factory Method solve:**

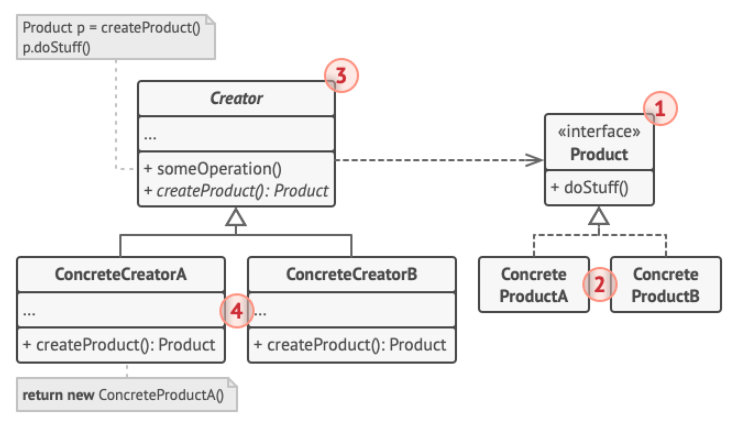
1. Move the object creation logic from our code to a separate class.
2. We use this pattern when we do not know in advance which class we may need to instantiate beforehand & also to allow new classes to be added to the system and handle their creation without affecting the client code.
3. Let subclasses decide which object to instantiate by overriding the factory method.

**Solution:**

1. Replace direct object construction calls with calls to a special factory method. Object returned by a factory method are often referred to as products



**UML:**



1. The Product declares the interface, which is common to all objects that can be produced by the creator and its subclasses.
2. Concrete Products are different implementations of the product interface.
3. The Creator class declares the factory method that returns new product objects.
4. Concrete Creators override the base factory method so it returns a different type of product.

**Implementation:**

1. Create an interface or an abstract class (Shape).
2. Create concrete classes (Circle, Square, etc.) that implement the interface or extend the abstract class (Shape).
3. Create a factory class (ShapeFactory) which creates an object from one of the concrete classes.