

# Factory Design Pattern

---

Blake Michalzik A01084417

## What is it?

---

Gang of Four definition: "Define an interface for creating an object, but you let the subclasses decide which class to instantiate. Factory method lets a class defer instantiation to subclasses."

The factory pattern takes the responsibility out of instantiating an object from the the class, instead it's handled by a Factory method class.

By using a factory method we are able to instantiate objects with a factory class dynamically.

This is typically done with classes that utilize the same interface.

Factory pattern makes our code less coupled, in turn making code more robust and easy to extend.

## Why use it?

---

A class is inflexible when using an object it creates directly. This is because it commits the class to a particular object and makes it impossible to change the instantiation independently of the class.

Changing the instantiator would require a change to the class code which, is bad practice. This is referred to as *code coupling* and the Factory method pattern assists in *decoupling* the code.

## Resources

---

[https://en.wikipedia.org/wiki/Factory\\_method\\_pattern](https://en.wikipedia.org/wiki/Factory_method_pattern)

<https://www.journaldev.com/1392/factory-design-pattern-in-java>