

Factory Patterns

Object Oriented System Design

Otago Polytechnic
Dunedin, New Zealand

DISCLAIMER

I hate factory patterns.

PARTIAL RETRACTION

I used to hate factory patterns because I saw them used pointlessly by “enterprise” programmers who slavishly used patterns without thinking.

This led to code that was hard to read and hard to reason about.

PATTERNS ARE JUST TOOLS

Our goal is always to write readable good code. Patterns are just a tool you can use when it helps with this goal.

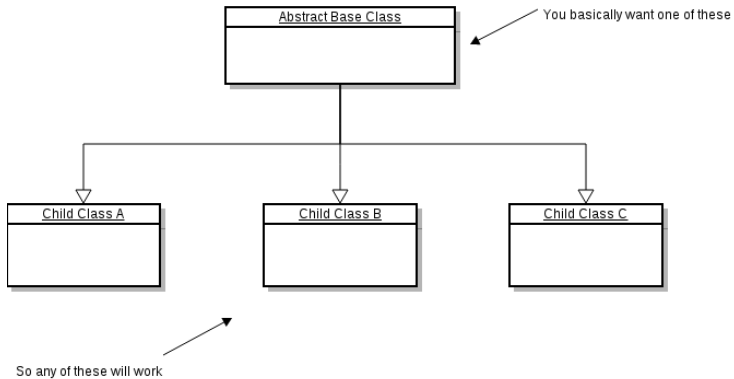
PROBLEM 0

- ▶ You're writing a game.
- ▶ In your game players fight zombies, vampires, and skeletons.
- ▶ At some point in the game you want to generate some monsters, but you don't know which type until runtime.

PROBLEM 1

- ▶ You're writing a document layout module.
- ▶ Your module includes a family of layout classes for various page dimensions (e.g., broadsheet, tabloid, poster board)
- ▶ Users will select which layout type to use at runtime.

THE PROBLEM, SIMPLY



SOLUTION: BUILD A FACTORY

```
class MonsterFactory:

    def create_monster(desired_type):
        if desired_type == 'vampire':
            return Vampire()
        elif desired_type == 'zombie':
            return Zombie()
        elif desired_type == 'skeleton':
            return Skeleton()
        else:
            return None
```


IS THAT IT?

- ▶ Sort of, yes.
- ▶ Ok, not really.
- ▶ We can implement various types of decision-making logic in our factories.
- ▶ Maybe we need to do a bit of housekeeping before we create a `Monster`.
- ▶ Maybe we need to create a parallel object at the same time we create a `Monster`, e.g., a `MonsterRenderer`.

EXERCISE

- ▶ Suppose we have data records in dictionaries with strings as keys.
- ▶ We want to represent these records as different sorts of documents:
 1. JSON
 2. YAML
 3. XML
- ▶ Write a set of Document classes, e.g., `JSONDocument` that implement a method, `dump()` that emits a data record in the correct document format.
- ▶ Then, write a `DocumentFactory` class that produces the desired type of document class.
- ▶ Experiment with different types of creation logic by implementing different create methods.
- ▶ Tips:
 1. Look for libraries to help you.
 2. Don't go nuts on the XML. If you can produce a crude XML structure that suffices.