Design Patterns

* Similar to mathematical formulas, we have design patterns in programming
* If formulas help you solve problems that follow a particular variable set and problem type, design patterns help with common programming design issues

Design Patterns we’ve used:

**Repository DP**: the BL shouldn’t be concerned with how the DL is storing stuff. Wrap the data access logic in a repository class that takes care of storage and retrieval of information.

**Façade DP**: wrapping a complex subsystem and presenting a simple interface that only contains the methods utilized by the end user.

**Design Patterns that you could use:**

**Singleton Design Pattern**:

Created a single instance (and only one instance) of a class and only utilizing that instance throughout program life cycle.

**Factory Design Pattern**:

This common superclass, create a factory class that produces certain implementations of that superclass depending on some input.

For example:

If you have menus that inherit from one menu interface, use a menu factory and use it to present a user a certain view depending on their input.