# The Big Picture on encapsulated behaviors…

* We’re using composition to give objects behaviors
* Those behaviors are encapsulations – separate, standalone, not dependent on other classes, like our Duck classes
* Instead of inheriting their behavior, Ducks get their behavior by being composed with the right behavior objects
* Favor composition over inheritance! HAS-A is better than IS-A.
* Lets you change behavior at runtime, without changing the source code
* Composition is used in many design patterns – it’s a fundamental building block

# Almost forgot … you just learned another new Design Pattern … The Strategy Pattern!