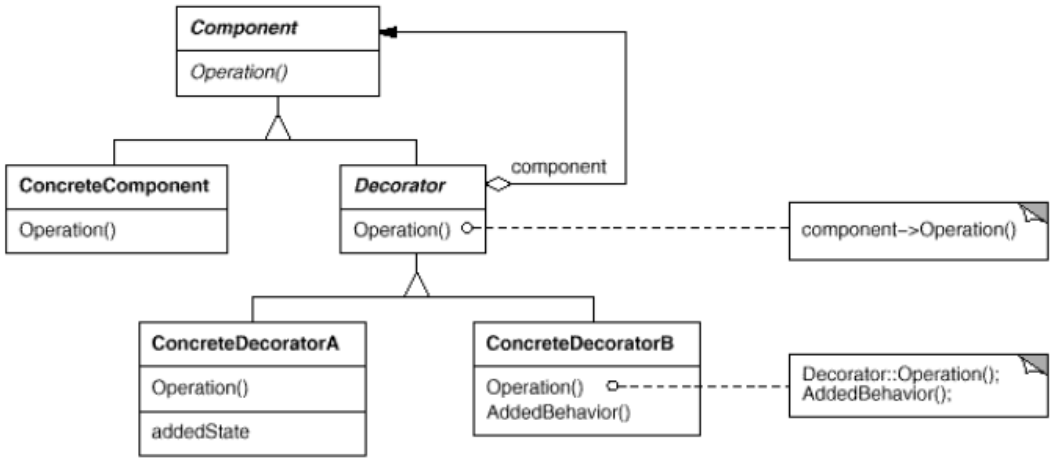
The **Decorator Pattern** is shown in the class diagram below.



1. As usual, think about what's going on with this pattern first.
2. We'll start by using this pattern to solve the following problem: Consider a simple system that sends out emails. When the email is sent to an external server, a "disclaimer" shall be added to the end of the original email. (e.g., from "eecs.wsu.edu" to ".gov").
   a) Draw a class diagram following the Decorator pattern.

   b) Write up pseudo code as well.

3. Consider the following questions, and make some general observations of the Decorator pattern.
    a) Benefits? Drawbacks?

    b) Is the abstract “Decorator” class necessary? Why/when would it (not) be?

    c) Compare to Composite: differences? And with that in mind, any new ideas on benefits and drawbacks?

    d) When should we use this, and when should we use Composite? Are they mutually exclusive?

4. Consider the Plants vs. Zombies example. How would the takeDamage() function work? What about adding a new functionality, that allows a magnet-mushroom to suck away the metal accessories immediately from the zombie?