**Goals:** To practice designing, implementing, and testing solutions using proper OO methodologies.

Automatic Transmissions

Cars with a manual transmission require the driver to change gears as they are changing speed. Putting it simply, this changes how much of the rotation of the engine affects the rotation of the axles (low gears transfer rotation slowly, so that even if the engine is revved up the car does not speed up much). This is the same as how gears on a bicycle work.

Cars with automatic transmissions solve this problem by letting the car computer choose how much rotation to transfer at certain speeds. Since cars cannot have negative speed (but can go as fast as their engine\'s allow), this choice of gear is a monotonically increasing function of speed.

In this lab, you will be simulating this in a computer program. Vehicles cannot have a negative speed, but can go as fast as needed.

What to do

**Package:** transmission

Design an interface called Transmission that represents a single car transmission. Then implement this interface in a class called AutomaticTransmission which automatically determines the current gear by the current speed of the car. Your implementation should include:

* A constructor that takes 5 speed thresholds for the 6 possible gears in order (speed to go from 1 to 2 or back, speed to go from 2 to 3 or back, etc.). The constructor should ensure that the input values are valid and throw an IllegalArgumentException if any of the parameter values are not legal.
* A method called increaseSpeed which returns a Transmission object with speed increased by 2 and the appropriate gear.
* A method called decreaseSpeed which returns a Transmission object with speed decreased by 2 and the appropriate gear. This method should throw an IllegalStateException if the speed becomes invalid.
* Methods that get the speed and the current gear. These are accessors (a.k.a., *getters*), so they should be named appropriately.
* A toString method that can be used to get the current state of the transmission (speed and gear), as follows:
* Transmission (speed = 45, gear = 3)

Transmission (speed = 0, gear = 0)

Testing

Whenever you write a class, you should also write tests for that class that proves not only that your code CAN work but that it WILL ALWAYS work. Your goal here is to write tests that achieve as close to 100% coverage as possible. In this class, we will measure coverage using Eclipse's "Coverage As ..." feature (right-click on the test/ directory to find this option).

Even more important than test coverage is that your tests should be written in such a way to ***convince someone else that your code works correctly.***