Day 06 – Design and Implement the Car Class

Abstraction is a representation, like a map, which captures the essentials

Other examples of abstractions?

- You can drive a car without knowing how the internal combustion engine works
- You can fly in a plane without understanding how the Bernoulli Principle works

When designing a ***class***, you can think of it as a ***category***:

- Human
- Professional Football team
- Grocery Store
- And the one we'll explore today, Automobile

Then, within that class, you can think of a particular ***instantiation*** or ***object*** of it:

- Human → you, your friend, …
- Professional Football team → Chicago Bears, Pittsburgh Steelers, Philadelphia Eagles, …
- Grocery Store → Giant, Weis, Wegmans, …
- Automobile → Toyota Highlander, Honda CRV, Ford Escape, …

In designing a Class, think of ***Attributes*** that are common for the particular Objects (Instantiations) of that class:

- Doors
- Tires
- Odometer
- Others?

Use Beginning Object-Orientation as a guide, two files, on separate panels on the board, Car.java and TDCar.java.

TD stands for ***Test Driver***. Creating the class is separate from using the class, two different files.

Design and implement your Car.java and TDCar.java
Iterated development!
Cite your collaborators and sources!