Encapsulation in the "Real" World

You might find encapsulation a strange idea; why make it harder for your programs to access their data? In fact, out in the real world, all of us are familiar with the ideas of encapsulation. Let me give you a few examples.

- **Today's automobiles** are more complex than Henry Ford's original car. Despite this, **driving** the latest Tesla is similar to, if not simpler than, driving a Model-T. Why, because, as cars got more complex, that complexity was **hidden** behind a **simpler** interface: the ignition (key), steering wheel, accelerator and brakes. These internal changes don't require drivers to take a new "how to drive" course. The **implementation details** are **encapsulated**.
- **Your computer** is another marvel of complexity. Unless you are a hardware tech, though, you never open up the system unit and try to operate it by shorting the circuits with a paper-clip. Instead, you use the **interface**—the mouse, and keyboard — to control the complex working parts that it contains.
- Finally, think about **your TV**. It's probably at least as complex as your car or your computer, but you don't need a license or a degree to operate one. Thanks to the magic of encapsulation, exemplified by the **remote control**, every child in the country can harness that power, although if you are a parent or grandparent, you might wish that were not true.

Just as with automobiles, computers and TVs, when it comes to programming, instead of making things more difficult, encapsulation makes objects safer **and** easier to use.

Encapsulation is one of the pillars underlying OOP or Object-Oriented Programming. We'll cover the other two, **inheritance** and **polymorphism**, next week.