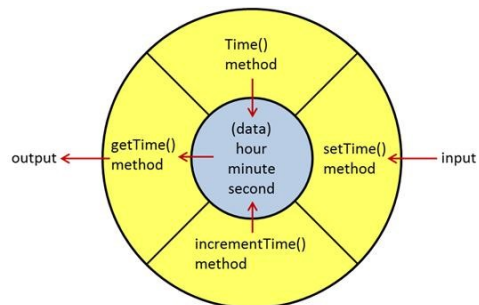


# What is Encapsulation?

With structures, the functions that make up the program, and the data the functions operate on, **are separate**. In an object-oriented program, **they are combined**. This process of wrapping up procedures and data together is called **encapsulation**.



Encapsulation is used to enforce the principle of **data hiding**, and, to allow your objects to **enforce their own invariants**, as we saw in the last chapter. With encapsulation, the data members defined inside a class are accessible to all the member functions defined inside the same class, but cannot be accessed by methods outside that class.

As you saw with the **Time** structure, making the data publicly accessible risks accidental data corruption as a result of a bug in someone else's code. The **struct** version of the **Time** type provides **no abstraction** and **no encapsulation**.

The **Time** **interface is its implementation**—the operations that clients can perform on the **Time** object are simply **direct manipulation** of its data. Changing the implementation thus **changes the interface**, which is why changing the data members breaks existing code.



This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.