Information Hiding

The `Time` structure from the last lesson did not enforce its invariants. Structures use "naked" variables to represent data, so **any part of the program** can modify those variables without any validation. **`Time` expects** certain relationships between its data members, **but cannot enforce those relationships**.

In addition, **`Time`** is **represented in a particular way**, as two **`int`** data members. Thus code that uses the **`Time`** type is **tightly coupled to that implementation**.

These problems with structured data were noticed in the early 1970s by a Canadian Computer Scientist named David Parnas, who developed a theory of **information hiding**, and who led the drive towards **modular programming** with his famous 1971 paper, *On the Criteria to be used in Decomposing Systems*, written at Carnegie Mellon University.