Accessors

Why should not use getter/setter methods

Problems relatived to Data abstraction
Changing Coding to change implement, since These functioms provide access to implement details in Class
- Causing deep trouble when changing a small implement
  - change type of field, and consequently lots of errors when calling it numerous times.
  - Modifying accessed field's type means you must change return value, also definition of that class, consequently, damage entire program.

Violating Encapsulation
- The presence of these methods may increase data movement
- system that heavily or inappropriately uses accessors simply isn't object oriented

When should using accessors
- a method to return an object in terms of an interface that the object implements
  - interface isolates you from changes to the implementing class.
- unnecessary flexibility is bad, but in these boundary APIs, the extra flexibility is unavoidable

Reason
- Without a clearly defined dynamic model, lead to design-by-guessing strategy.
- designs by force of habit

Solution
- Using user interface (UI) construction
- the CRC (classes, responsibilities, collaboration) card (a teaching methodology)(