

OOP – Data encapsulation

The concept of encapsulation

Encapsulation is a way of organizing data and methods into a structure by concealing the way the object is implemented, i.e. preventing access to data by any means other than those specified. Encapsulation therefore guarantees the integrity of the data contained in the object.

Concealing data

The user of a particular class does not need to know how the data in that object is structured, this means that a user does not need to know how implementation takes place. By preventing the user from directly modifying attributes, and forcing the user to use defined functions in order to modify them (called **interfaces**), data integrity is thus ensured (for example, one can ensure that the data type given matches expectations, or is returned within the expected time interval).

Encapsulation defines the access levels for elements of that class. These access levels define the access rights to the data, allowing us to access the data by a method of that particular class itself, from an inheritance class, or even from any other class. There are three levels of access:

- **public**: functions of all classes may access the data or methods of a class that is defined with the *public* access level. This is the lowest level of data protection
- **protected**: data access is restricted to functions of inheritance classes, i.e. member functions of that class and all sub-classes

- **private:** data access is restricted to methods of that particular class only. This is the highest level of data protection

Source: <http://en.kioskea.net/contents/421-oop-data-encapsulation>