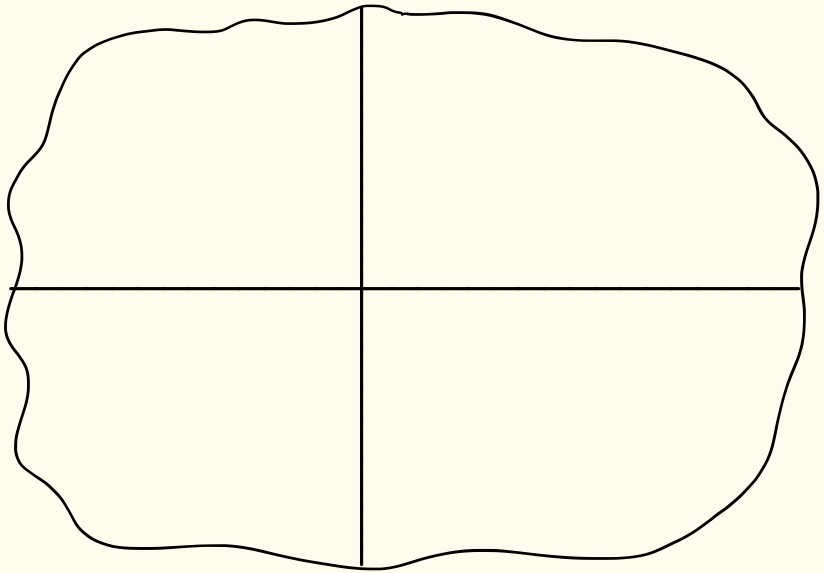
Software Life Cycle

1. Specification
2. Design (algorithms and data structures)
3. Implementation (translate design to programming language)
4. Testing and debugging

Design

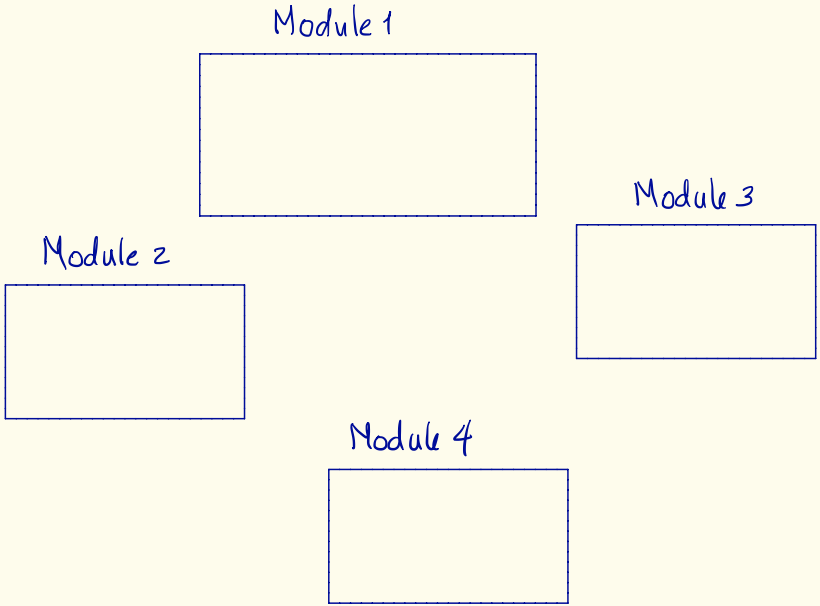
- To solve a complex problem we divide it into simpler, smaller problems.
- We solve each smaller problem.
- We combine the solutions of the smaller problems to get the solution to the original, complex problem.



Modular Design

A program designed in this manner will consist of several parts, or modules, each one solving one of the smaller problems.

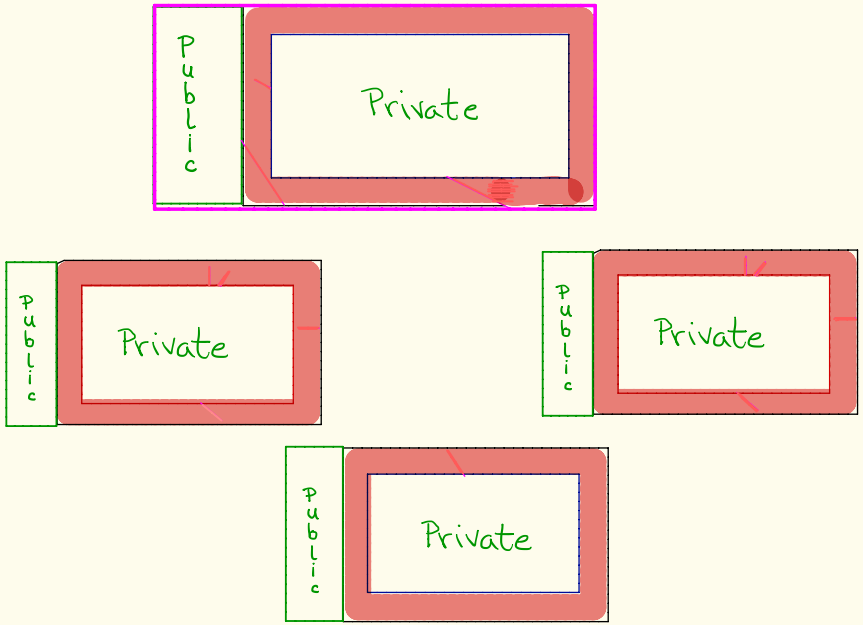
We wish the modules to be as independent from each other as possible; this simplifies their construction and in a team environment allows each module to be assign to a different member of the team. Lack of module dependences allows team members to work independently.



Modules

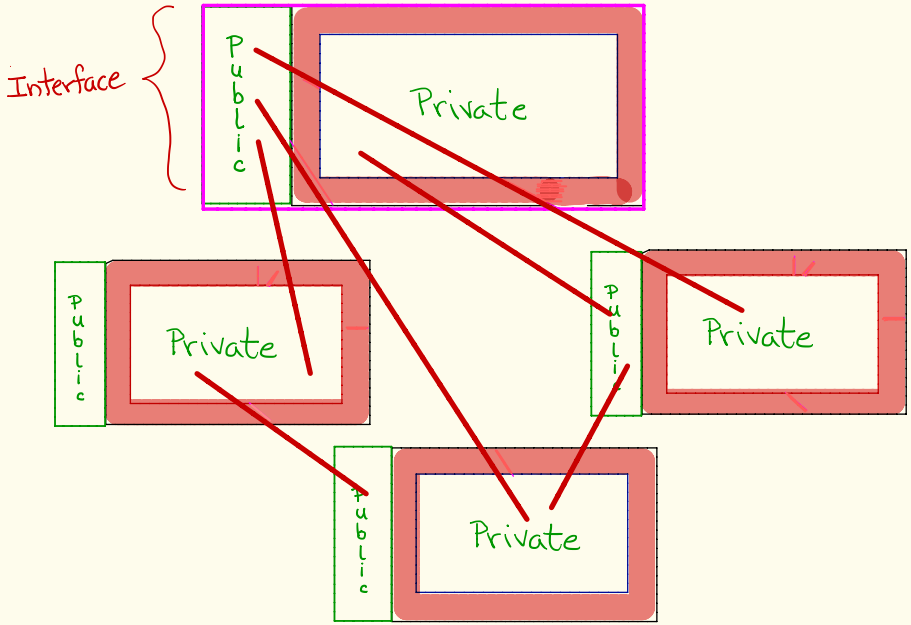
Each module contains data and code. Part of a module is kept hidden from other modules to achieve independence.

However, since the modules need to collaborate to solve the whole problem, part of a module is public or available to other modules.



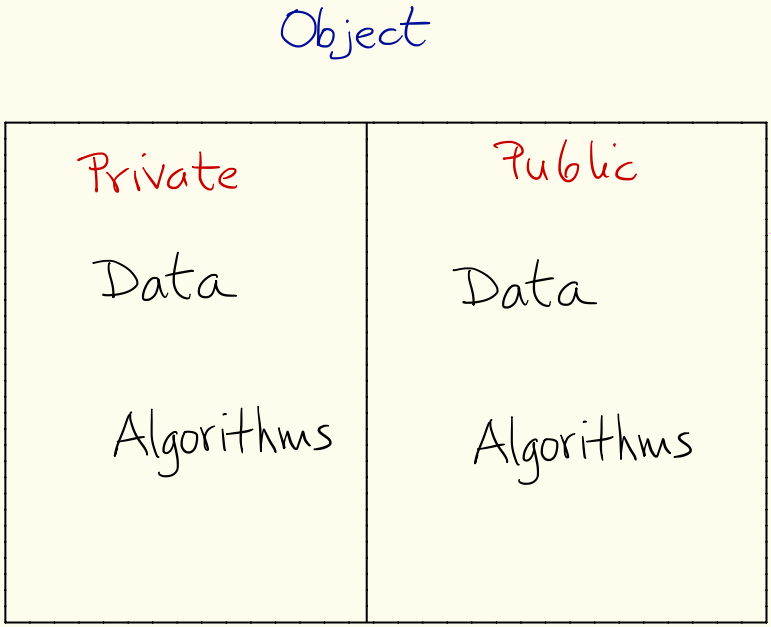
Module Interaction

Interaction among the modules takes place through their public part or interfaces.



Objects

In Object Oriented (OO) terminology a module is called an object.



In general it is not a good programming practice to make data public, as then it can be manipulated in arbitrary (and sometimes unexpected) ways