SOFTWARE DESIGN & ANALYSIS (Week-6)

USAMA MUSHARAF

MS-CS (Software Engineering)

LECTURER (Department of Computer Science)

FAST-NUCES PESHAWAR

CONTENTS OF WEEK # 6

- Design Principles and Concepts
- Assignment # 1 & 2

DESIGN PRINCIPLES

DESIGN PRINCIPLES

1- The design process should not suffer from “tunnel vision.”

A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job, and the design concepts.

2- The design should be traceable to the analysis model.

DESIGN PRINCIPLES

3- The design should not reinvent the wheel.

- Systems are constructed using a set of design patterns.
- These patterns should always be chosen as an alternative to reinvention.
- Time is short and resources are limited!

DESIGN PRINCIPLES

4- The design should “minimize the intellectual distance” between the software and the problem as it exists in the real world.

DESIGN PRINCIPLES

5- The design should be structured to accommodate change

6- The design should be assessed for quality as it is being created

DESIGN CONCEPTS

FUNDAMENTAL CONCEPTS OF DESIGN

- **abstraction**—data, procedure, control
- **refinement**—elaboration of detail for all abstractions
- **modularity**—compartmentalization of data and function
- **architecture**—overall structure of the software
  - Styles and patterns
- **procedure**—the algorithms that achieve function
- **hiding**—controlled interfaces

ABSTRACTION

"Capture only those details about an object that are relevant to current perspective"

Suppose we want to implement abstraction for the following statement,

"Ali is a PhD student and teaches BS students"

*Here object Ali has two **perspectives** one is his **student perspective** and second is his **teacher perspective.***

ABSTRACTION

A cat can be viewed with different perspectives.

Ordinary Perspective	Surgeon's Perspective
A pet animal with	A being with
Four Legs	A Skeleton
A Tail	Heart
Two Ears	Kidney
Sharp Teeth	Stomach

ABSTRACTION

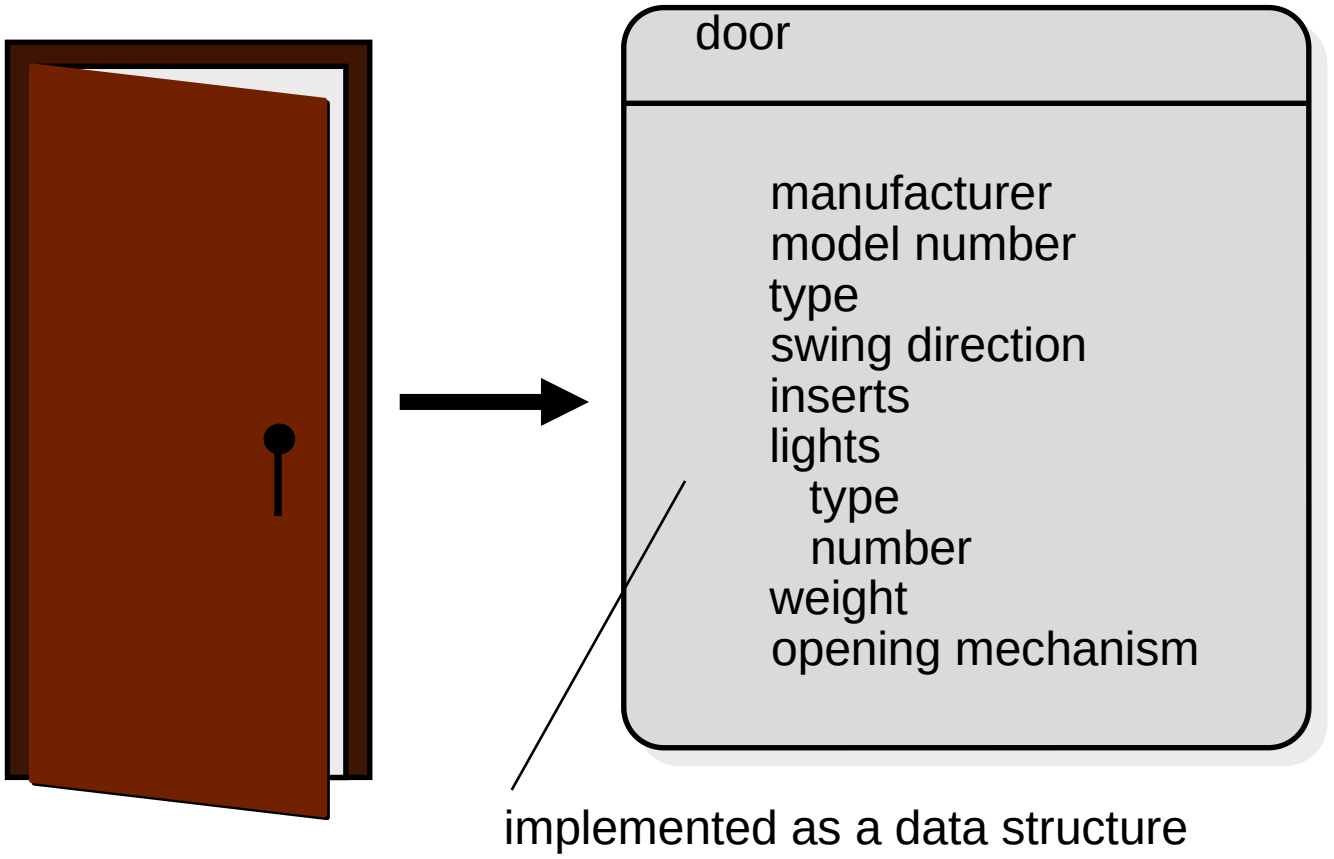
Driver's View

Engineer's View

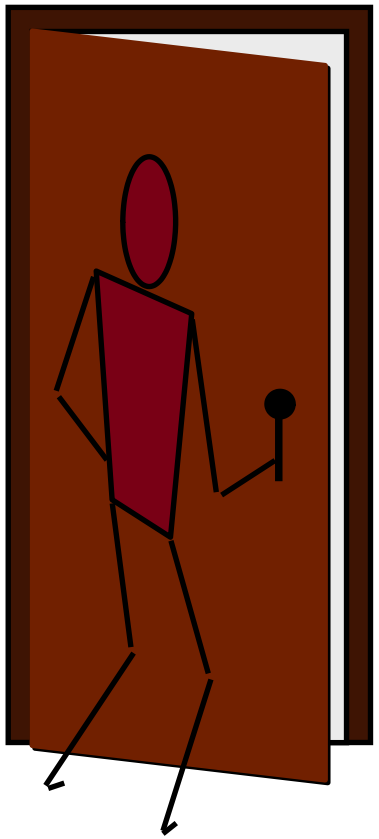
ABSTRACTION ADVANTAGES

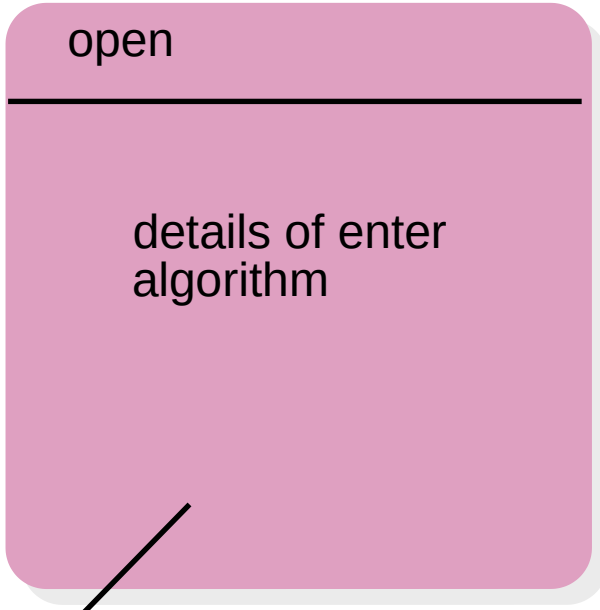
- It helps us understanding and solving a problem using object oriented approach as it hides extra irrelevant details of objects.
- Focusing on single perspective of an object provides us freedom to change implementation for other aspects of for an object later.
- *Abstraction is used for achieving information hiding as we show only relevant details to related objects, and hide other details.*

DATA ABSTRACTION



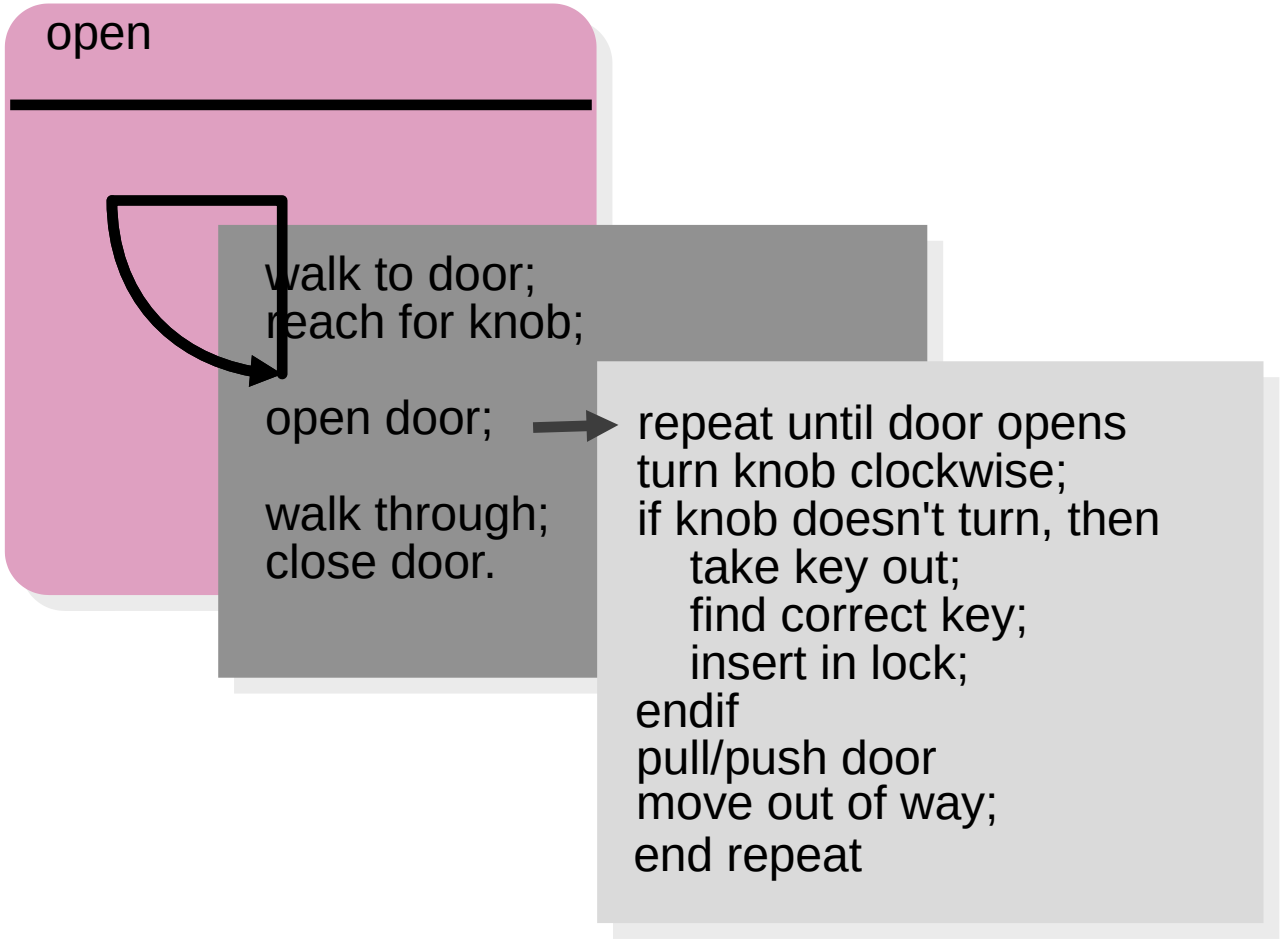
PROCEDURAL ABSTRACTION





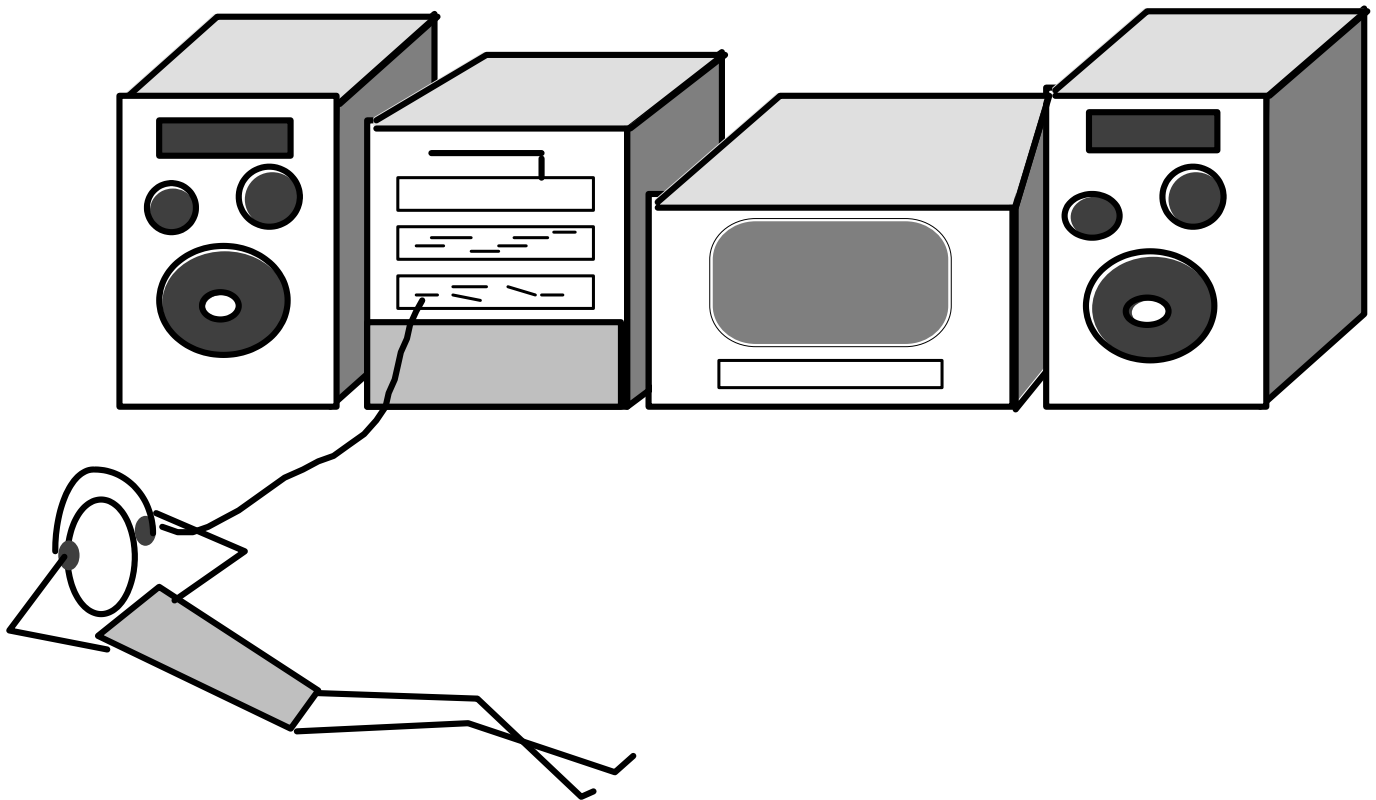
implemented with a "knowledge" of the object that is associated with enter

STEPWISE REFINEMENT



MODULAR DESIGN

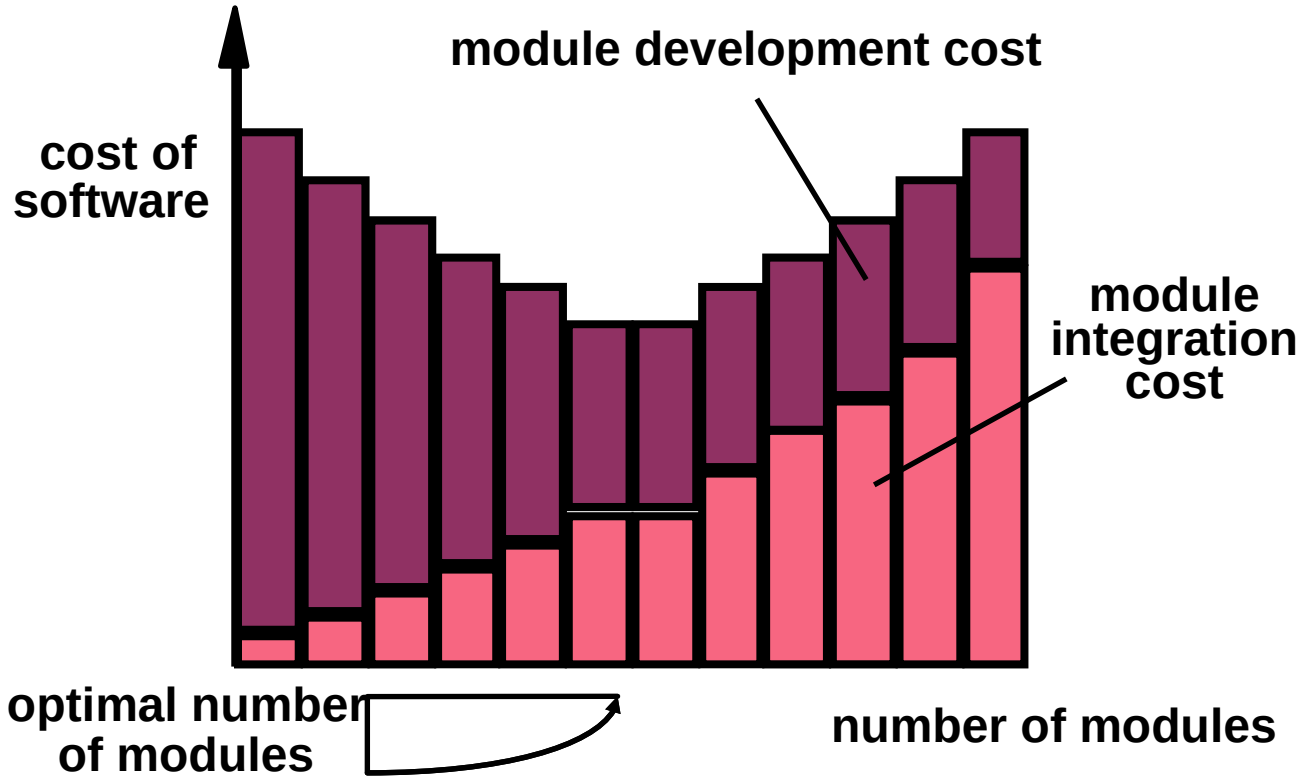
easier to build, easier to change, easier to fix …



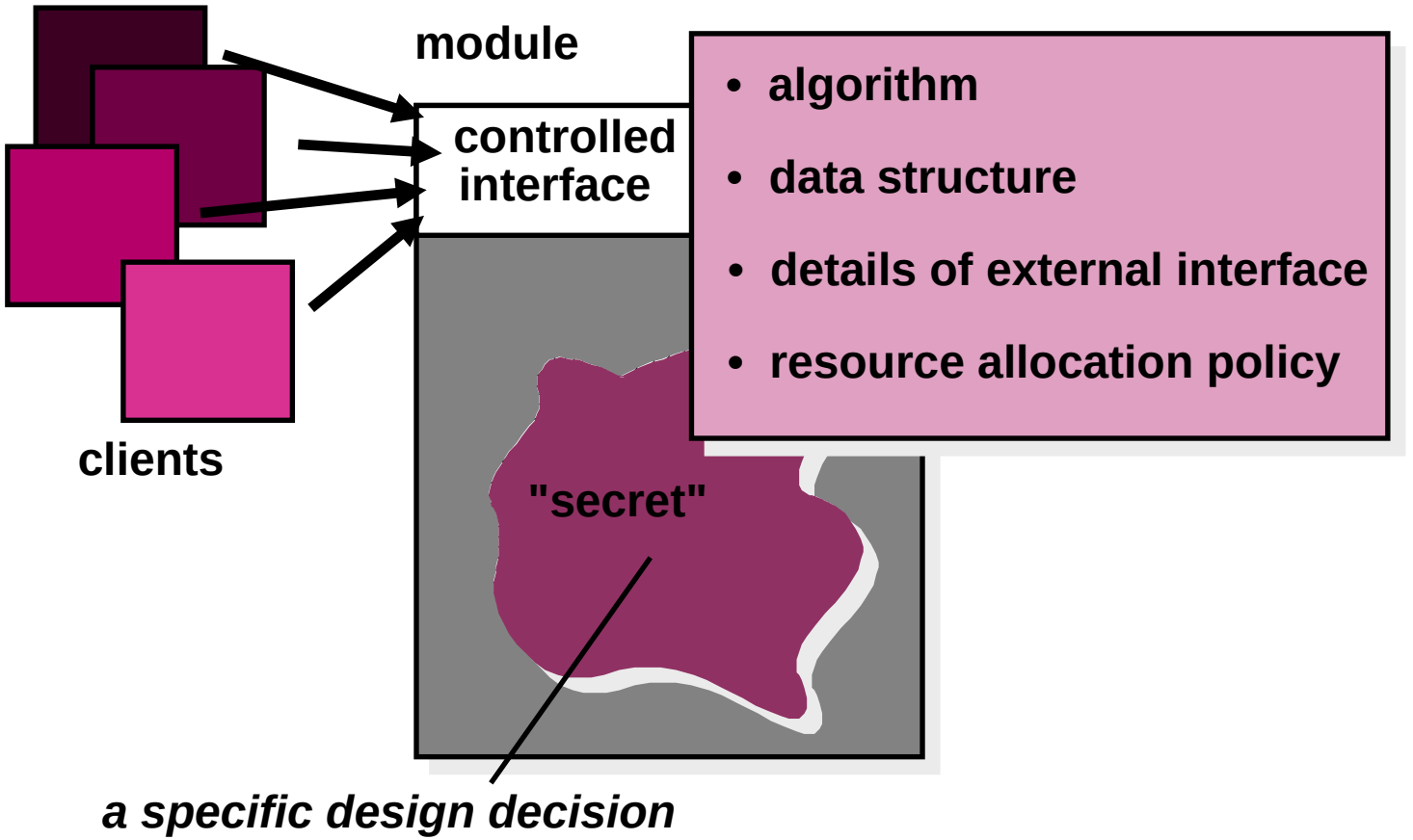
- **Easier to manage**
- **Easier to understand**
- **Reduces complexity**
- **Delegation / division of work**
- **Fault isolation**
- **Independent development**
- **Separation of concerns**
- **Reuse**

MODULARITY: TRADE-OFFS

What is the "right" number of modules for a specific software design?



INFORMATION HIDING



INFORMATION HIDING

- Design the modules in such a way that information (data & procedures) contained in one module is inaccessible to other modules that have no need for such information.
- Independent modules.

Benefits:

when modifications are required, it reduces the chances of propagating to other modules.

EFFECTIVE MODULAR DESIGN

FUNCTIONAL INDEPENDENCE

COHESION - the degree to which a module performs one and only one function.

COUPLING - the degree to which a module is "connected" to other modules in the system.

COUPLING

Coupling is a measure of independence of a module or component.

Loose coupling means that different system components have loose or less reliance upon each other.

Hence, changes in one component would have a limited affect on other components.

COUPLING

High coupling causes problems

- Change propagation- ripple effect
- Difficulty in understanding
- Difficult reuse

COHESION

Cohesion is a measure of the degree to which the elements of the module are functionally related.

It is the degree to which all elements directed towards performing a single task are contained in the component.

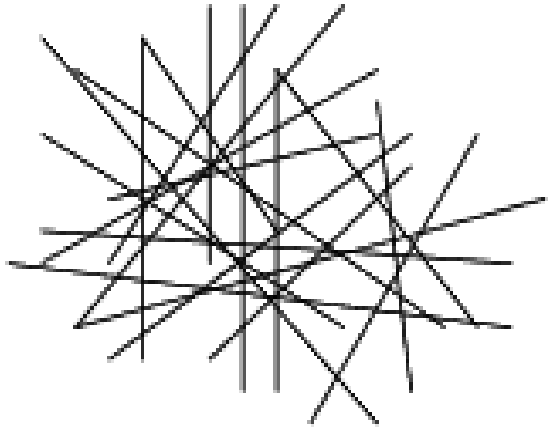
Basically, cohesion is the internal glue that keeps the module together.

A good software design will have high cohesion

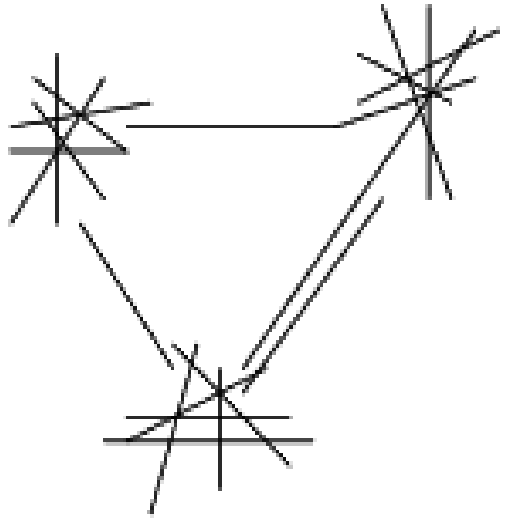
COUPLING & COHESION

A Software should be Lesly coupled and highly cohesive.

COUPLING

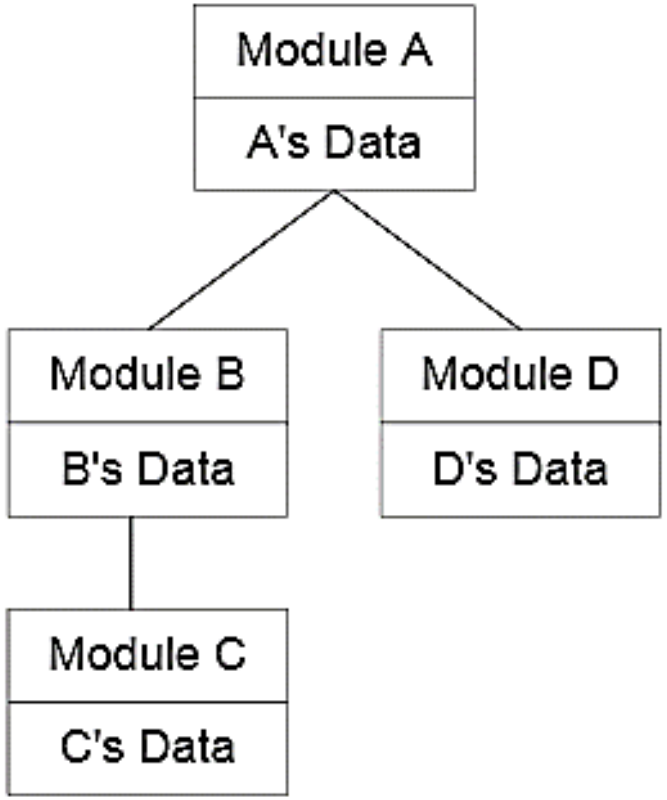


High Coupling

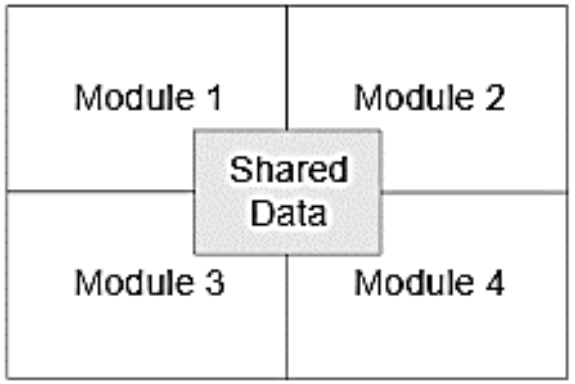


Low Coupling

COUPLING

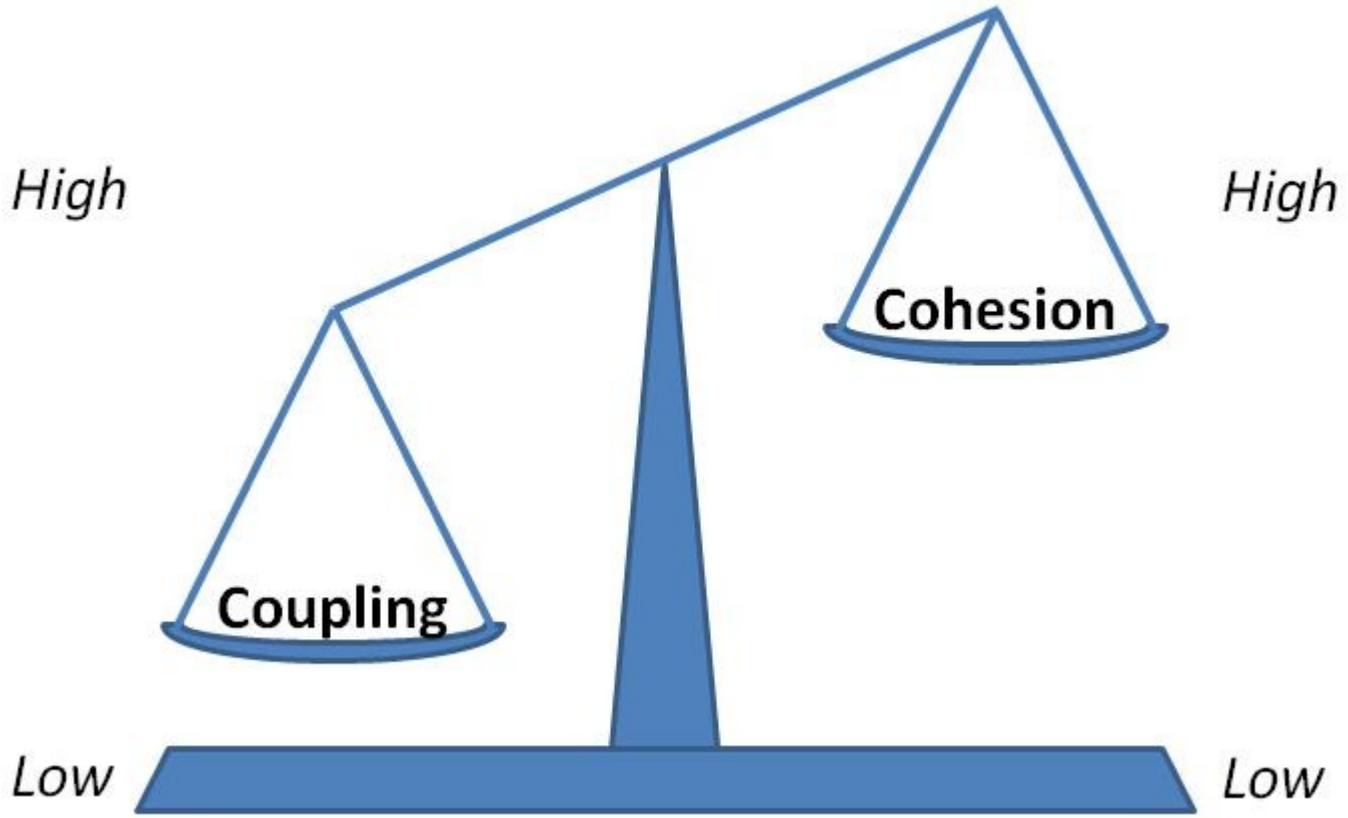


Low Coupling



High Coupling

RELATIONSHIP BETWEEN COUPLING AND COHESION



HAVE A GOOD DAY!