SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Course Name – Software Engineering

II Year / III Semester

DESIGN CONCEPTS AND PRINCIPLES

Modular Design

Modular Design

- Modular design reduces the design complexity and results in easier and faster implementation by allowing parallel development of various parts of a system.
- We discuss a different section of modular design in detail in this section:

1. Functional Independence:

- Functional independence is achieved by developing functions that perform only one kind of task and do not excessively interact with other modules.
- Independence is important because it makes implementation more accessible and faster.
- The independent modules are easier to maintain, test, and reduce error propagation and can be reused in other programs as well.
- Thus, functional independence is a good design feature which ensures software quality.

. **It is measured using two criteria:**

•**Cohesion:** It measures the relative function strength of a module.

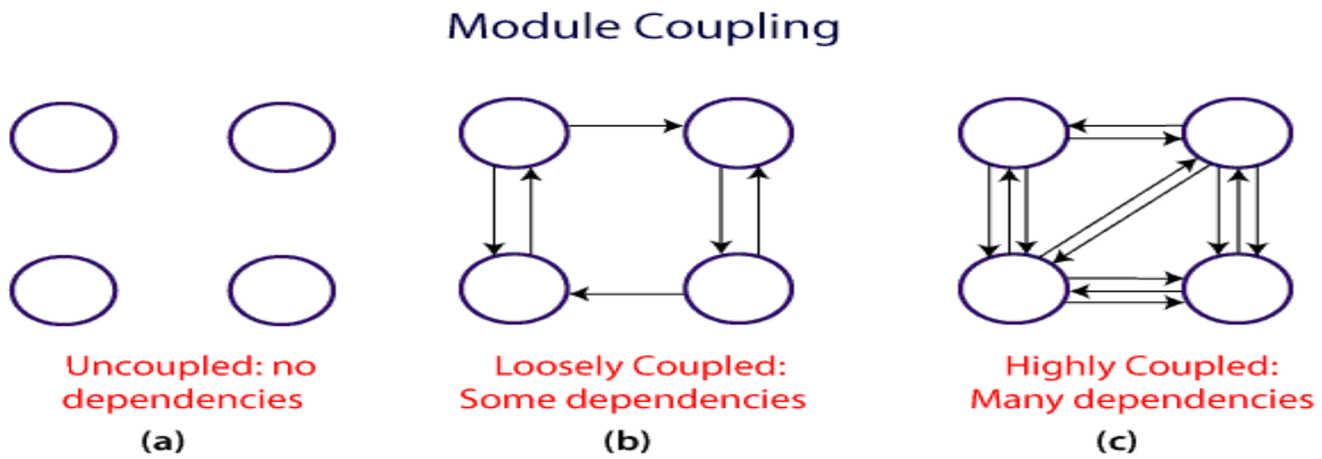
•**Coupling:** It measures the relative interdependence among modules.

2. Information hiding:

- The fundamental of Information hiding suggests that modules can be characterized by the design decisions that protect from the others, i.e., In other words, modules should be specified that data include within a module is inaccessible to other modules that do not need for such information.
- The use of information hiding as design criteria for modular system provides the most significant benefits when modifications are required during testing's and later during software maintenance.
- This is because as most data and procedures are hidden from other parts of the software, inadvertent errors introduced during modifications are less likely to propagate to different locations within the software.

Module Coupling

- In software engineering, the coupling is the degree of interdependence between software modules.
- Two modules that are tightly coupled are strongly dependent on each other.
- However, two modules that are loosely coupled are not dependent on each other.
- **Uncoupled modules** have no interdependence at all within them.



- A good design is the one that has low coupling.
- Coupling is measured by the number of relations between the modules.
- That is, the coupling increases as the number of calls between modules increase or the amount of shared data is large.
- Thus, it can be said that a design with high coupling will have more errors.

Module Cohesion

- In computer programming, cohesion defines to the degree to which the elements of a module belong together.
- Thus, cohesion measures the strength of relationships between pieces of functionality within a given module.
- For example, in highly cohesive systems, functionality is strongly related.
- Cohesion is an **ordinal** type of measurement and is generally described as "high cohesion" or "low cohesion."

Differentiate between Coupling and Cohesion

Coupling	Cohesion
Coupling is also called Inter-Module Binding.	Cohesion is also called Intra-Module Binding.
Coupling shows the relationships between modules.	Cohesion shows the relationship within the module.
Coupling shows the relative **independence** between the modules.	Cohesion shows the module's relative **functional** strength.
While creating, you should aim for low coupling, i.e., dependency among modules should be less.	While creating you should aim for high cohesion, i.e., a cohesive component/ module focuses on a single function (i.e., single-mindedness) with little interaction with other modules of the system.
In coupling, modules are linked to the other modules.	In cohesion, the module focuses on a single thing.

THANK YOU