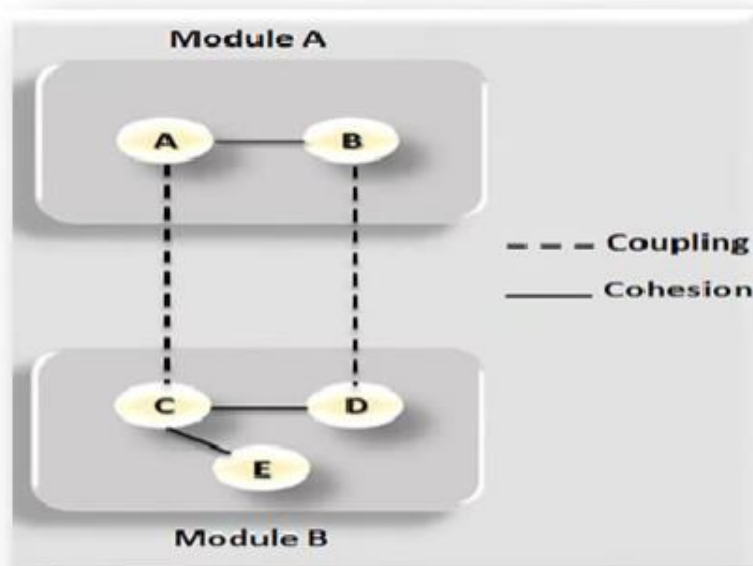


Software Engineering

Coupling & Cohesion

(Explain Types with Examples)



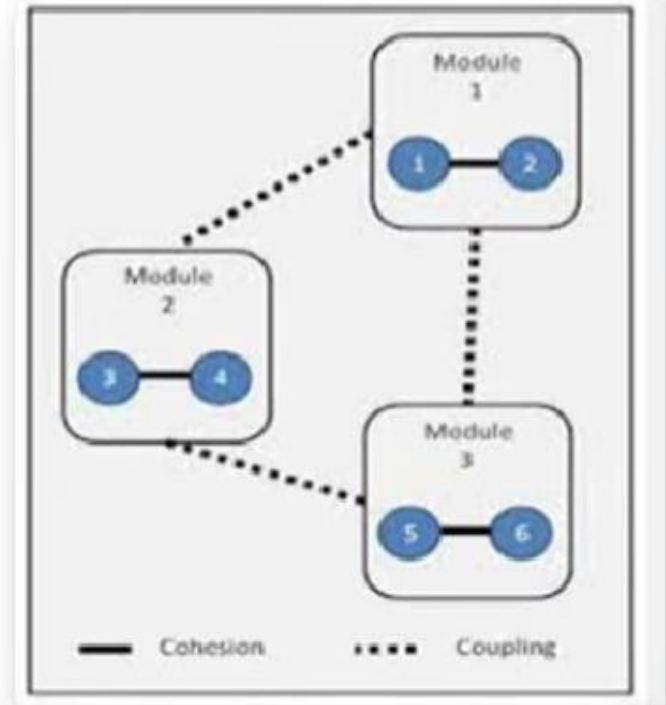
Ms. Vrushali Dhanokar

M.Tech CSE

dhanokarvrushali@gmail.com

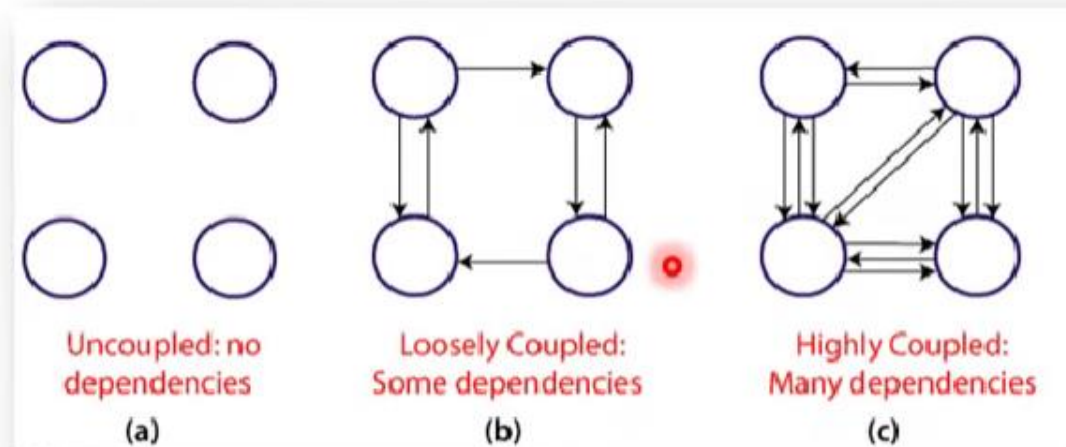
Contents

1. About Coupling
2. Types of Coupling
3. About Cohesion
4. Types of Cohesion
5. Good & Bad Software Design
6. Important Questions

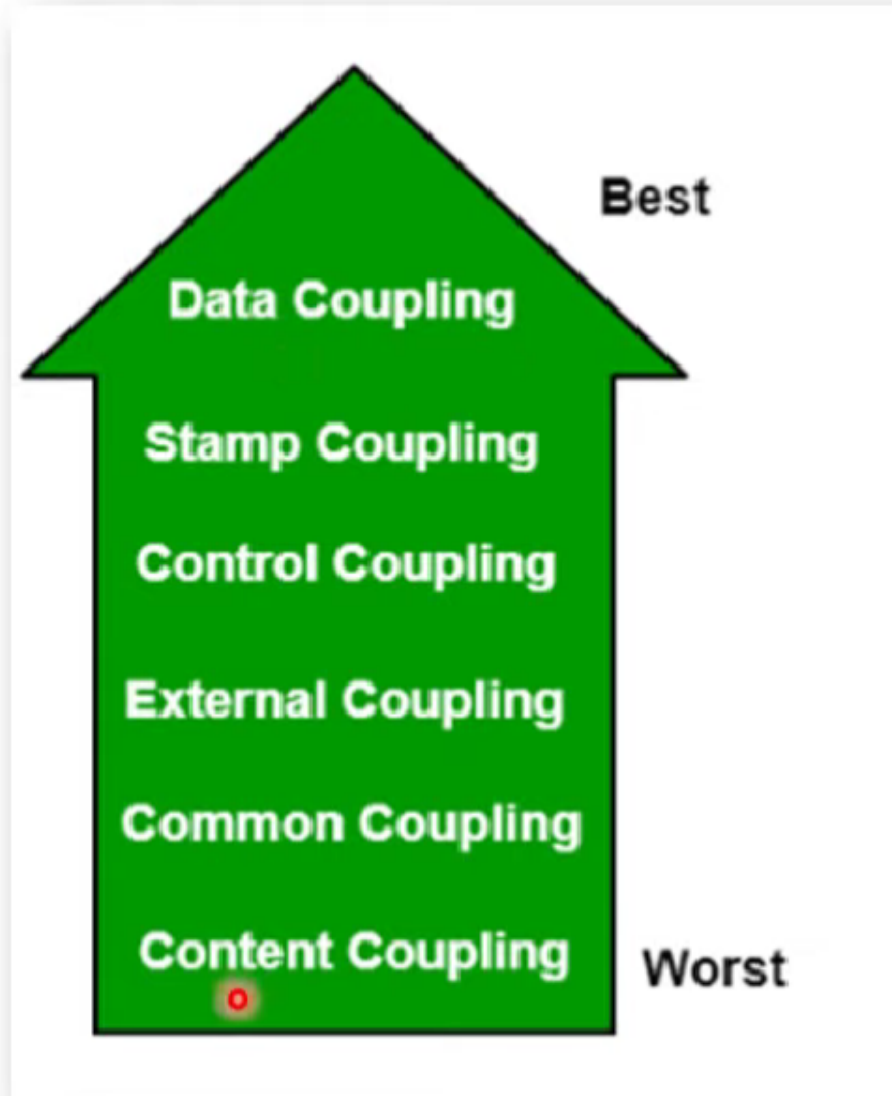


About Coupling

- The coupling is the degree of interdependence or number of relations between software modules.
- Two modules that are tightly coupled are strongly dependent on each other.
- However, two modules that are loosely coupled are not much dependent on each other.
- A **good design** is the one that has Low coupling.
- High coupling generates more errors because they shared large number of data.



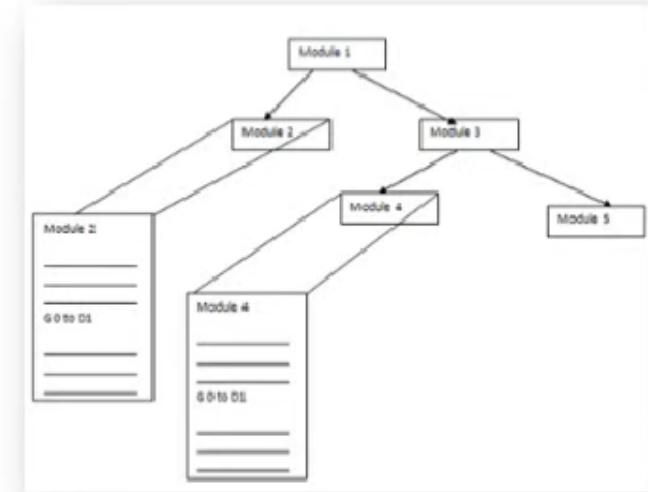
Types of Coupling



Types of Coupling

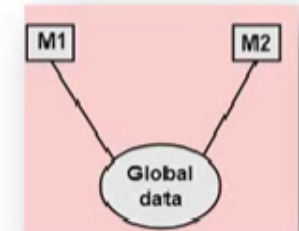
Type 1: Content Coupling

- Here, Two modules are connected as they share the same content like functions, methods.
- When a change is made in one module the other module needs to be updated as well.



Type 2: Common Coupling

- Two modules are common coupled if they share information through some global data items.



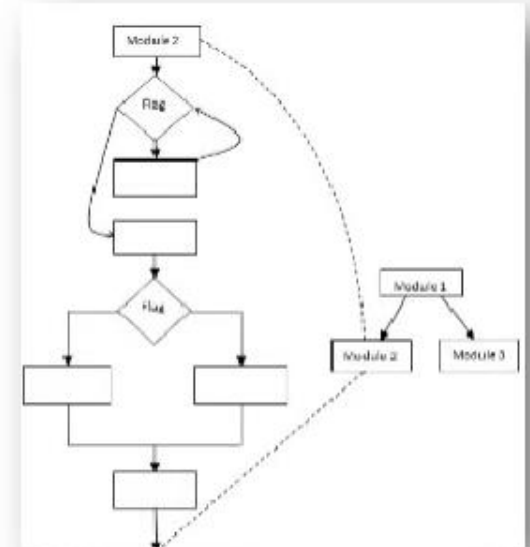
Types of Coupling

Type 3: External Coupling

- When two modules share an externally import data format, communication protocols or device interface.
- This is related to communication to external tools and devices.

Type 4: Control Coupling

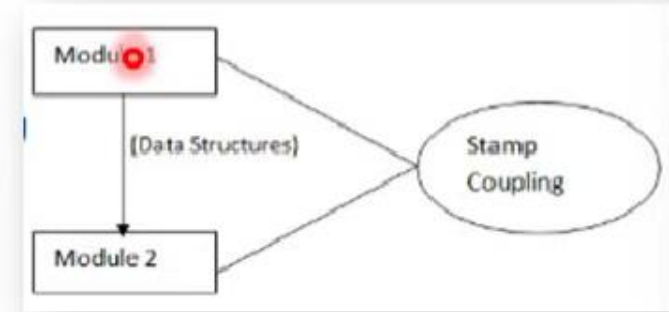
- Control coupling handle functional flow between software modules.
- **Example:** Module 1- Set Flag = 1 then only Module 2 perform action.



Types of Coupling

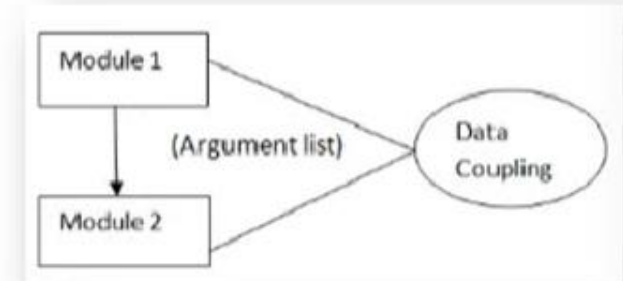
Type 5: Stamp Coupling

- Two modules are stamp coupled if they communicate using composite data items such as Complete Data Structure & objects.
- No junk or unused data shared between the two coupling modules.



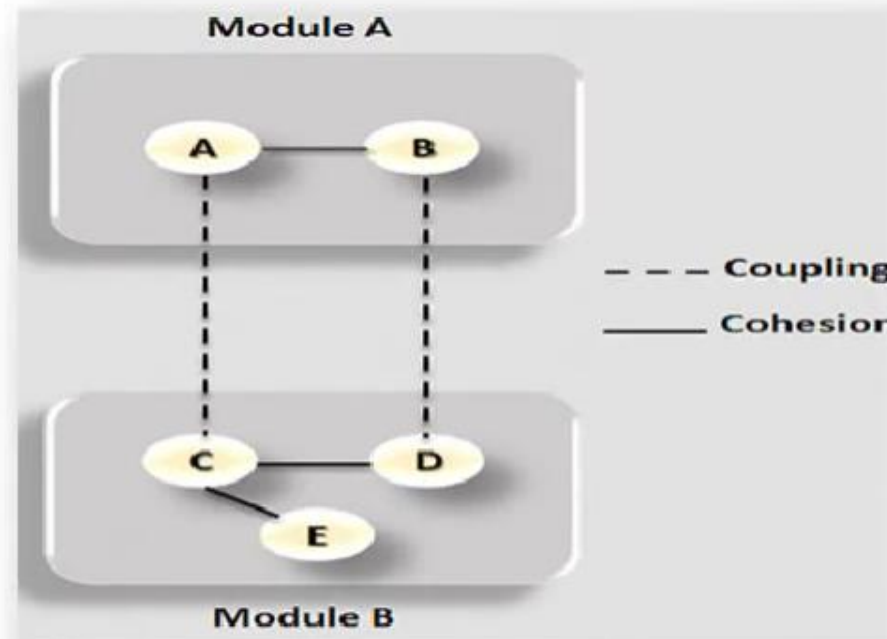
Type 6: Data Coupling

- When data are passed from one modules to another module via argument list or parameters through functional blocks.

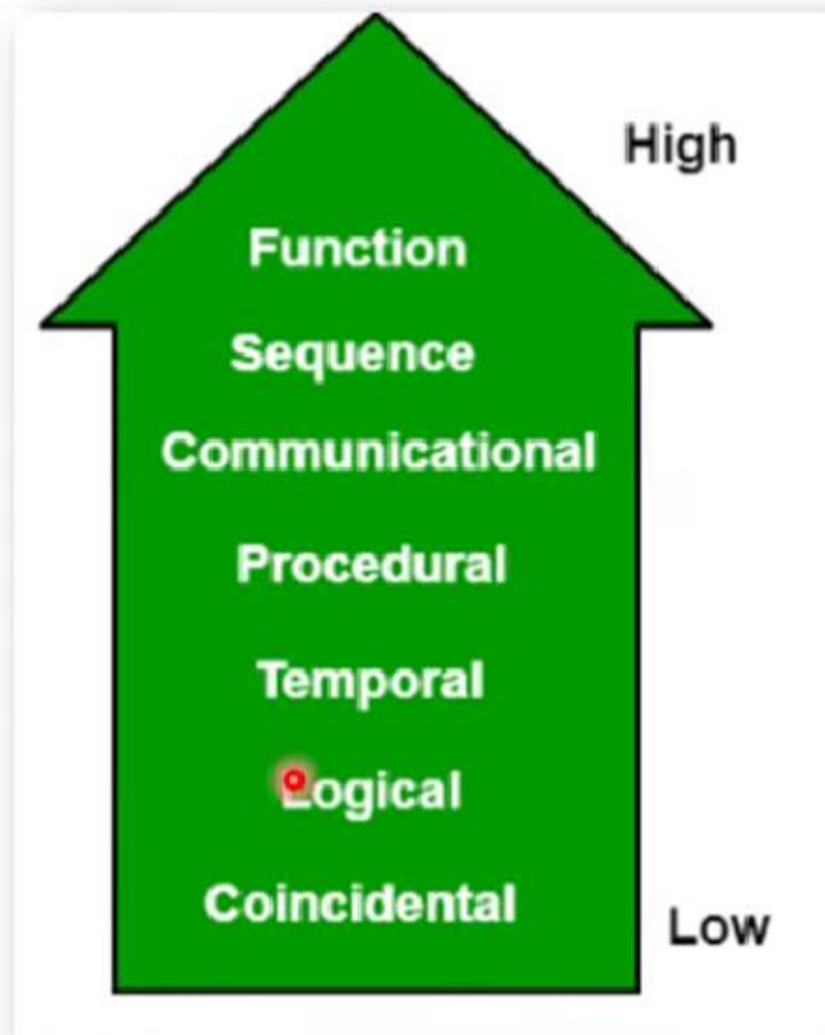


About Cohesion

- Cohesion defines to the degree to which the elements of a module belong together or interrelated.
- Thus, cohesion measures the strength of relationships between pieces of functionality within a given module.
- A **good software design** will have high cohesion.



Types of Cohesion



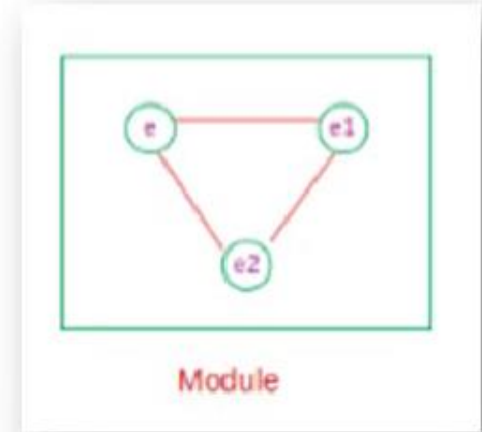
Types of Cohesion

Type 1: Coincidental Cohesion

- It performs a set of tasks that are associated with each other very loosely.
- **Example:** Calculator : ADD, SUB, MUL, DIV

Type 2: Logical Cohesion

- If all the elements of the module perform a similar operation.
- **Example:** Error handling, Sorting, If Type of Record = Student then Display Student Record.



Type 3: Temporal Cohesion

- The activities related in time, Where all methods executed at same time.
- Temporal cohesion is found in the modules of initialization and termination.
- **Example:** Counter = 0, Open student file, Clear(), Initializing the array etc.

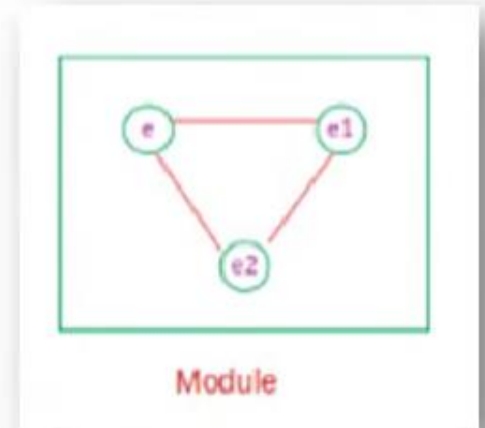
Types of Cohesion

Type 4: Procedural Cohesion

- All parts of a procedure execute in particular sequence of steps for achieving goal.
- **Example:** Calling one function to another function, Loop statements, Reading record etc.

Type 5: Communicational Cohesion

- If all the elements of a module are working on the same input & output data and are accessing that data through the same data structures.
- **Example:** Update record in the database and send it to the printer.



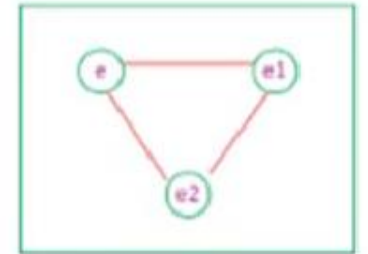
Types of Cohesion

Type 6: Sequence Cohesion

- Output of one element treats as an input to the other elements inside the same module.
- **Example:** Enter the numbers -> Perform Addition of that numbers -> Display Addition.

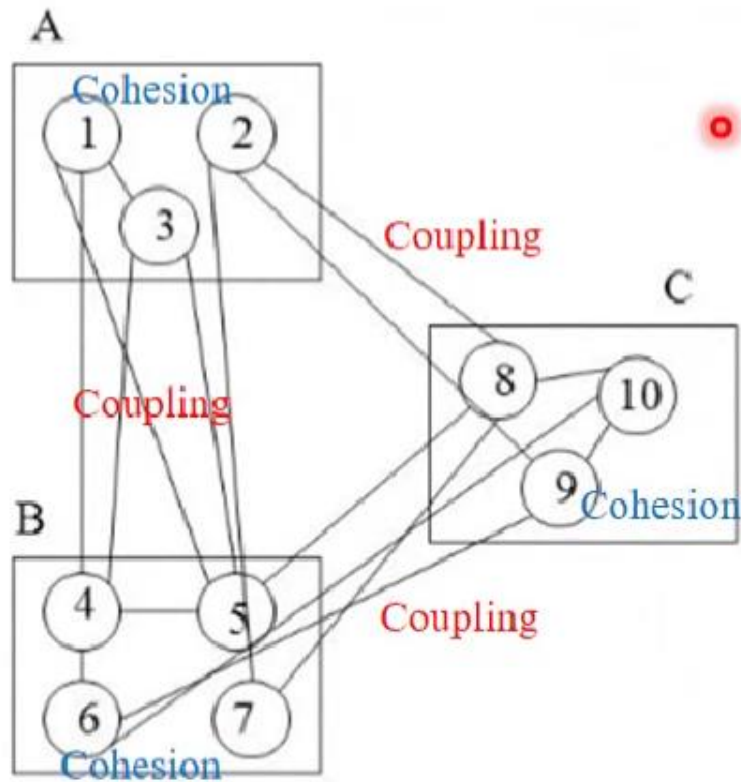
Type 7: Function Cohesion

- If a single module aims to perform all the similar types of functionalities through its different elements.
- The purpose of functional cohesion is single minded, high, strong and focused.
- **Example:** Railway Reservation System

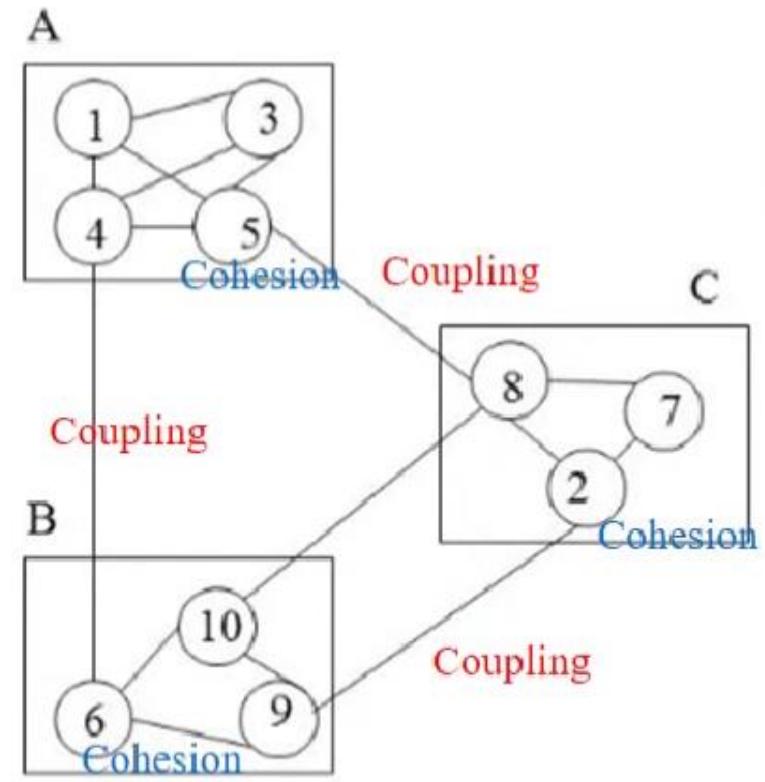


Module

Good & Bad Software System Design



Bad modularization:
low cohesion, high coupling



Good modularization:
high cohesion, low coupling

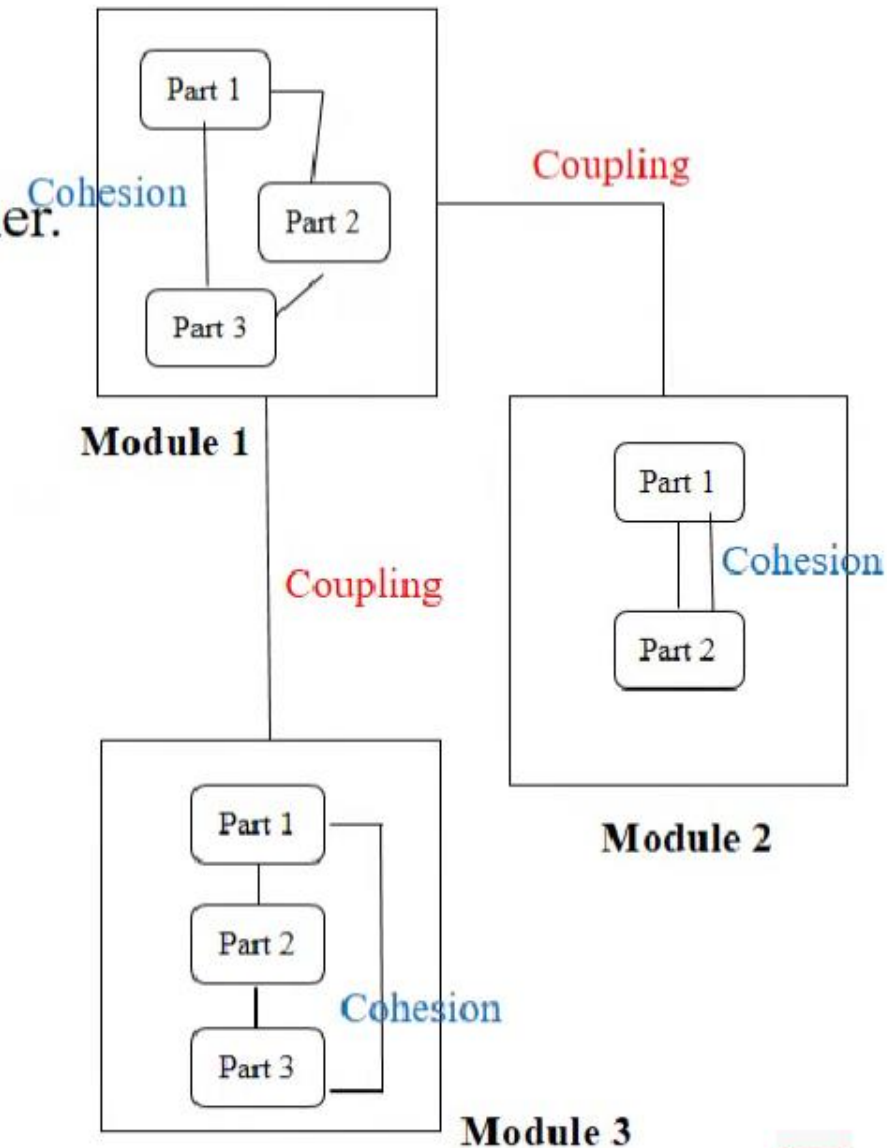
Why High Cohesive & Low Coupling generate good design?

➤ Due to Low Coupling

- **Readability:** Modules are easy to understand not complex.
- **Maintainability:** Changes in one module little impact on other.
- **Modularity:** Enhance modules development.
- **Scalability:** Adding new module remove existing one easy.
- **Testability:** Modules are easy to test & debug.

➤ Due to High Cohesion

- **Readability:** Related functions easy to understand.
- **Reusability:** Easily Reuse module in another system.
- **Reliability:** Generate overall improvement of system.
- **Testability:** Modules are easy to test & debug.



Important Questions

As per previous year question papers

1. What do you mean by term cohesion & coupling in the context of software design? How are the concepts are useful for good design system. 7M
2. Why software design should have high cohesive and low coupling? Justify. 8M



Thank You...!



LIKE



COMMENT



SHARE



SUBSCRIBE



THANKS FOR WATCHING