

12 Lecture - CS504

Important Subjective

Explain the concept of coupling in software design. Answer: Coupling refers to the degree of interdependence between software components. It measures how much one component relies on another. Low coupling is desirable as it promotes modularity, reusability, and flexibility.

What is cohesion in software design? Answer: Cohesion refers to the degree of relatedness of responsibilities within a software component. It measures how closely the functions or operations in a module are aligned with a single purpose. High cohesion is desirable as it promotes maintainability and code organization.

Discuss the difference between loose coupling and tight coupling. Answer: Loose coupling implies minimal dependencies between components, allowing them to be modified independently. In contrast, tight coupling indicates strong dependencies, where changes in one component may have a significant impact on others. Loose coupling promotes flexibility and ease of maintenance.

How does low coupling contribute to software design? Answer: Low coupling reduces dependencies between components, making the system more modular, flexible, and maintainable. It allows for independent development and easier testing and facilitates the introduction of changes without causing cascading effects.

What are the benefits of high cohesion in software design? Answer: High cohesion ensures that responsibilities within a component are closely related, making the component easier to understand, modify, and maintain. It promotes code reusability, reduces complexity, and improves overall system performance.

Explain the concept of data coupling. Answer: Data coupling occurs when components share data through parameters or data structures. It indicates a low level of interdependence and is considered desirable as it reduces the impact of changes in one component on others.

How can you improve cohesion in a software system? Answer: Cohesion can be improved by organizing functions and operations within modules based on their relatedness. Grouping related functions together and separating unrelated functions into separate modules can increase cohesion.

Discuss the challenges of tight coupling in software design. Answer: Tight coupling increases the interdependence between components, making the system less flexible and more difficult to modify or maintain. Changes in one component may require modifications in multiple other components, increasing the risk of introducing errors.

What is the relationship between coupling and dependency? Answer: Coupling refers to the degree of interdependence between components, while dependency represents a relationship where one component relies on another. Higher levels of coupling often result in stronger dependencies between components.

How can you reduce coupling in a software system? Answer: Coupling can be reduced by using well-defined interfaces, applying design patterns like Dependency Injection, employing loose coupling principles like the Dependency Inversion Principle, and modularizing the system into cohesive components.