Software Design Process - Exam Questions and Answers

1. What is the design process in software engineering?

The design process is an iterative and creative process of defining how to implement all customer requirements. It involves early decisions about system architecture and later decisions about unit-level implementation.

2. Why is software design considered a challenging task?

Software design is challenging due to the need to accommodate many possibilities, meet non-functional goals (e.g., maintainability), and adhere to external factors like standards and regulations.

3. What are design patterns and how do they aid the design process?

Design patterns are generic solutions for making lower-level design decisions. They help by offering proven approaches that solve recurring design problems.

4. List and briefly explain the overall goals of good software design.

Goals include increasing profit (reducing cost, increasing revenue), accommodating requirements, speeding up development, and improving qualities like usability, efficiency, reliability, maintainability, and reusability.

5. Explain the 'Divide and Conquer' design principle.

This principle involves breaking a large system into smaller, manageable parts. It simplifies understanding, allows component specialization, and enables easier modification and replacement of parts.

6. What is cohesion and why is it important in software design?

Cohesion measures how closely related the responsibilities of a module are. High cohesion makes systems easier to understand, maintain, and reuse.

7. Define and differentiate the types of cohesion: Functional, Layer, Communicational,

Sequential, Procedural, Temporal, and Utility.

Functional: single computation task.

Layer: grouped by related services.

Communicational: grouped by data access.

Sequential: output from one is input to another.

Procedural: procedures executed in order.

Temporal: same phase execution.

Utility: generic helpers used across systems.

8. What is coupling and how does it affect software systems?

Coupling is the degree of interdependence between modules. High coupling leads to difficulties in understanding and maintaining systems. Low coupling is desirable.

9. Describe content coupling and how to avoid it.

Content coupling occurs when one module modifies the internal data of another. It can be avoided by encapsulating data (using private variables and accessors).

10. Explain control coupling and a strategy to reduce it.

Control coupling happens when a method controls another by passing a command/flag. It can be reduced using polymorphism or a lookup table mapping commands to methods.