PG-DESD AUG 2024

Module Name: Software Design

Note: This question bank is only for your reference. first, go through the videos, then do these questions.

1. What is a key aspect of General Design Concepts?
   A. Ignoring usability principles
   B. Understanding the separation of concerns
   C. Introducing complex dependencies
   D. Hardcoding business logic
   Answer: B

2. What is a primary concern in the Context of Software Design?
   A. Data encryption techniques
   B. Balancing stakeholder requirements and technical constraints
   C. Implementing low-level code
   D. Ignoring scalability requirements
   Answer: B

3. Which activity is included in the Software Design Process?
   A. Creating machine-level instructions
   B. Defining software architecture and components
   C. Documenting only testing strategies
   D. Implementing UI designs without a plan
   Answer: B

4. What is the purpose of Software Design Principles?
   A. To make software designs more rigid and unchangeable
   B. To provide guidelines for creating maintainable, modular, and extensible designs
   C. To reduce readability of code
   D. To focus on hardware constraints
   Answer: B

5. Which of the following is a Software Design Principle?
   A. Minimize cohesion
   B. Maximize coupling
   C. Encapsulation
   D. Use of magic numbers
   Answer: C

6. What does the term ‘Separation of Concerns’ refer to?
   A. Handling all concerns in a single module
   B. Dividing a software system into distinct features
   C. Ignoring data management
   D. Focusing on hardware control
   Answer: B

7. Which principle encourages using well-defined interfaces between modules?
   A. Minimize code reuse

B. Encapsulation
C. Overcomplication
D. Redundancy
Answer: B

8. What does the software design principle of ‘Open/Closed’ mean?
A. Software entities should be open for extension, but closed for modification
B. Always allow modifications to any class
C. Restrict the addition of new functionalities
D. Ignore the software lifecycle
Answer: A

9. What is an example of Context in Software Design?
A. Hardware constraints
B. Platform dependencies
C. User requirements
D. All of the above
Answer: D

10. Which of the following describes a Good Design?
A. High cohesion and low coupling
B. Low cohesion and high coupling
C. Ignoring interdependencies
D. Minimizing module functionality
Answer: A

11. What does Concurrency refer to in software design?
A. Running multiple processes in sequence
B. Executing multiple processes simultaneously
C. Writing unstructured code
D. Ignoring race conditions
Answer: B

12. What is the purpose of Control and Handling of Events?
A. To capture and respond to user inputs and system events
B. To disable user interactions
C. To introduce delays
D. To implement only synchronous interactions
Answer: A

13. Which of the following deals with managing data between sessions?
A. Control Handling
B. Data Persistence
C. Event Propagation
D. Dependency Injection
Answer: B

14. What is a common method for ensuring Data Persistence?
A. Using variables only in RAM
B. Storing data in databases or files
C. Ignoring session management

D. Using temporary files

Answer: B

15. What is the focus of Distribution of Components?
    A. Keeping all components in a single location
    B. Distributing software components across different machines or networks
    C. Ignoring service interactions
    D. Minimizing scalability

    Answer: B

16. What is a key consideration in Error and Exception Handling?
    A. Propagating errors without handling
    B. Implementing fault tolerance mechanisms
    C. Increasing system downtime
    D. Ignoring error messages

    Answer: B

17. What is the purpose of Fault Tolerance?
    A. Making the system fragile
    B. Ensuring the system continues operating even in the presence of faults
    C. Increasing system failures
    D. Ignoring resilience

    Answer: B

18. Which of the following enhances Interaction and Presentation?
    A. Ignoring user experience
    B. Following usability guidelines for designing the user interface
    C. Using a plain-text user interface
    D. Overloading UI components

    Answer: B

19. What is an important aspect of Security in Software Design?
    A. Using weak passwords
    B. Implementing robust authentication and authorization mechanisms
    C. Ignoring encryption
    D. Storing sensitive data in plain text

    Answer: B

20. Which design issue deals with managing simultaneous data access?
    A. Control Handling
    B. Concurrency
    C. Event Propagation
    D. Data Obfuscation

    Answer: B

21. What does an Architectural Structure define?
    A. The physical layout of a database
    B. The organization and interactions of system components
    C. The visual aesthetics of the software
    D. The syntax of a programming language

    Answer: B

22. Which viewpoint focuses on non-functional requirements like performance and scalability?
    A. Logical View
    B. Physical View
    C. Deployment View
    D. Security View
    Answer: C
23. Which of the following is an example of an Architectural Style?
    A. Layered Architecture
    B. Magic Numbers
    C. Overcomplication
    D. Disconnected Modules
    Answer: A

24. What is the purpose of a Design Pattern?
    A. To introduce complexity
    B. To provide a reusable solution for common design problems
    C. To increase code redundancy
    D. To minimize code reusability
    Answer: B

25. Which pattern is used to provide a simplified interface to a complex subsystem?
    A. Singleton
    B. Facade
    C. Observer
    D. Factory
    Answer: B

26. What is a benefit of using Design Patterns?
    A. Ignoring best practices
    B. Enforcing a proven solution for common issues
    C. Minimizing code readability
    D. Making code harder to understand
    Answer: B

27. Which of the following is a type of Architectural Style?
    A. Observer
    B. MVC (Model-View-Controller)
    C. Factory
    D. Singleton
    Answer: B

28. What is a primary factor in Architecture Design Decisions?
    A. Cost and Resource Constraints
    B. Ignoring security
    C. Overloading the system with features
    D. Avoiding modularity
    Answer: A

29. Which of the following is a common Family of Programs and Frameworks?
    A. Object Pool Pattern

B. .NET Framework
C. Using Magic Numbers
D. Ignoring standards
Answer: B

30. What is a Framework?
A. A rigid design that cannot be extended
B. A reusable, semi-complete application structure
C. A set of low-level hardware instructions
D. A code snippet repository
Answer: B

31. What is an example of a General Design Concept?
A. Hardcoding dependencies
B. Defining layers and modules to separate functionalities
C. Ignoring design constraints
D. Using global variables
Answer: B

32. Which of the following is included in the Context of Software Design?
A. Business requirements, technical constraints, and external dependencies
B. Ignoring project deadlines
C. Focusing only on UI elements
D. Disregarding user requirements
Answer: A

33. Which stage in the Software Design Process focuses on breaking down a system into smaller components?
A. Integration
B. Architectural design
C. Code optimization
D. Documentation
Answer: B

34. Which design principle emphasizes reducing the impact of changes?
A. High coupling
B. Minimize cohesion
C. Information hiding
D. Ignoring modularity
Answer: C

35. What is meant by ‘Single Responsibility Principle’?
A. A class should have only one reason to change
B. A class should handle multiple responsibilities
C. A function should perform unrelated tasks
D. A module should be independent of others
Answer: A

36. What is the significance of ‘Don’t Repeat Yourself (DRY)’ in software design?
A. To create redundancy
B. To eliminate duplication and improve maintainability

C. To introduce dependencies
D. To minimize performance
Answer: B

37. Which Software Design Principle focuses on defining object interactions?
A. Loose coupling
B. Tight coupling
C. Using monolithic structures
D. Ignoring modularity
Answer: A

38. What does ‘Separation of Concerns’ aim to achieve?
A. Keeping different aspects of the system isolated for independent development
B. Merging all concerns into a single module
C. Ignoring scalability
D. Overloading class responsibilities
Answer: A

39. What is a common technique to handle Concurrency in software design?
A. Blocking all processes
B. Using locks and semaphores
C. Disabling multi-threading
D. Avoiding resource sharing
Answer: B

40. What is the primary goal of Control and Handling of Events?
A. Ignoring real-time constraints
B. Efficiently managing and responding to different system and user events
C. Introducing race conditions
D. Reducing system responsiveness
Answer: B

41. What is the role of a Transaction Manager in Data Persistence?
A. To ensure atomicity and consistency of data operations
B. To introduce data redundancy
C. To reduce data security
D. To minimize data storage
Answer: A

42. Which of the following deals with distributing components across different platforms?
A. Control Handling
B. Platform Independence
C. Distribution of Components
D. Concurrency Management
Answer: C

43. What is a common method used in Error and Exception Handling?
A. Try-catch blocks
B. Ignoring exceptions
C. Failing silently

D. Propagating all errors without handling

Answer: A

44. What is an example of Interaction and Presentation design?
    A. Data encryption techniques
    B. Using appropriate UI elements to guide the user
    C. Creating database schemas
    D. Implementing low-level code

    Answer: B

45. Which of the following enhances Security in Software Design?
    A. Hardcoding passwords
    B. Implementing input validation and output encoding
    C. Storing sensitive data in plain text
    D. Avoiding role-based access control

    Answer: B

46. Which of the following is a common design issue in Concurrency?
    A. Deadlock
    B. Data Persistence
    C. Control Handling
    D. Exception Propagation

    Answer: A

47. What is an advantage of using Layered Architecture?
    A. Mixing unrelated functionalities
    B. Promoting separation of concerns by organizing system components into layers
    C. Hardcoding dependencies
    D. Ignoring system scalability

    Answer: B

48. Which Architectural Style is commonly used in distributed systems?
    A. Monolithic
    B. Microservices
    C. Coupled Modules
    D. Single-tier

    Answer: B

49. What is a benefit of using Design Patterns like Observer?
    A. Coupling components tightly
    B. Allowing multiple components to react to changes in one component without tight coupling
    C. Increasing system dependencies
    D. Minimizing system flexibility

    Answer: B

50. What does a Framework provide in Software Design?
    A. A concrete implementation with no flexibility
    B. A reusable design that can be extended for various applications
    C. A low-level syntax checker
    D. A rigid structure with minimal use cases

Answer: B

51. Which design principle helps in promoting code reusability and maintenance?
    A. Cohesion
    B. Single Responsibility Principle
    C. Tight Coupling
    D. Global Access

    Answer: B

52. What does ‘Open-Closed Principle’ state?
    A. Software entities should be open for extension, but closed for modification
    B. Software entities should be closed for both extension and modification
    C. Classes should be open for modification
    D. Functions should be closed for extension

    Answer: A

53. Which term refers to the capability of different objects to be accessed through a common interface?
    A. Inheritance
    B. Polymorphism
    C. Aggregation
    D. Encapsulation

    Answer: B

54. Which concurrency issue involves multiple processes waiting for each other to release resources?
    A. Starvation
    B. Race Condition
    C. Deadlock
    D. Buffer Overflow

    Answer: C

55. What is a key advantage of using Event-Driven Architecture?
    A. Sequential processing
    B. Parallel execution of unrelated tasks
    C. Ignoring user events
    D. Blocking system resources

    Answer: B

56. Which design concept ensures that internal details of a module are hidden from others?
    A. Coupling
    B. Encapsulation
    C. Inheritance
    D. Polymorphism

    Answer: B

57. Which tool is commonly used for managing and handling concurrency?
    A. Debugger
    B. Profiler
    C. Mutex
    D. Compiler

Answer: C

58. What is a primary characteristic of a Microservices Architecture?
    A. Single database for all components
    B. Small, independent services working together
    C. Monolithic structure
    D. Global state sharing

    Answer: B

59. What does ‘Dependency Inversion Principle’ emphasize?
    A. High-level modules should not depend on low-level modules
    B. Modules should be independent of each other
    C. Only low-level modules should have access to each other
    D. Ignoring dependencies altogether

    Answer: A

60. Which type of error handling involves setting up checkpoints to recover from failures?
    A. Defensive Programming
    B. Fault Tolerance
    C. Error Ignoring
    D. Input Validation

    Answer: B

61. What is a common strategy for improving the Performance of a software system?
    A. Using caching mechanisms
    B. Avoiding modularity
    C. Hardcoding values
    D. Disabling memory optimizations

    Answer: A

62. What is the purpose of a Middleware in distributed software?
    A. To enable communication and data management between distributed components
    B. To handle user authentication
    C. To store configuration settings
    D. To disable system logs

    Answer: A

63. Which method is effective in constructing Heterogeneous Systems?
    A. Using cross-platform libraries
    B. Relying only on specific OS functions
    C. Ignoring compatibility issues
    D. Using platform-dependent APIs

    Answer: A

64. Which architecture pattern is based on dividing software into reusable components?
    A. Monolithic Architecture
    B. Component-Based Architecture
    C. Layered Architecture
    D. Service-Oriented Architecture

    Answer: B

65. What does the term ‘State-Based Construction’ refer to?
    A. Designing systems with specific states and transitions
    B. Hardcoding each system state
    C. Ignoring system states
    D. Designing without considering system behavior

    Answer: A

66. Which type of error occurs when two or more processes access shared resources simultaneously?
    A. Deadlock
    B. Race Condition
    C. Data Persistence
    D. Middleware Issue

    Answer: B

67. Which pattern is commonly used to provide a simplified interface to a complex system?
    A. Observer Pattern
    B. Facade Pattern
    C. Singleton Pattern
    D. Proxy Pattern

    Answer: B

68. What is the benefit of using Design by Contract?
    A. Enforcing clear preconditions, postconditions, and invariants
    B. Ignoring code dependencies
    C. Reducing maintainability
    D. Focusing only on performance

    Answer: A

69. What is the goal of ‘Test-First Programming’?
    A. Writing tests before implementing functionalities
    B. Ignoring test cases
    C. Creating documentation before testing
    D. Writing test cases only for completed modules

    Answer: A

70. Which term refers to the logical separation of components in software design?
    A. Layered Architecture
    B. Functional Programming
    C. Imperative Design
    D. Procedural Programming

    Answer: A

71. Which of the following patterns restricts the instantiation of a class to one object?
    A. Adapter Pattern
    B. Singleton Pattern
    C. Decorator Pattern
    D. Factory Pattern

    Answer: B

72. Which design principle suggests that a class should not have more than one reason to change?

A. Single Responsibility Principle
B. Interface Segregation
C. Dependency Inversion
D. Liskov Substitution
Answer: A

73. Which approach is useful in parameterizing similar code with generic parameters?
A. Generics
B. Tight Coupling
C. Static Binding
D. Overloading
Answer: A

74. What is the primary purpose of Profiling in Software Development?
A. Identifying performance bottlenecks
B. Removing redundant comments
C. Ignoring code quality
D. Performing syntax analysis
Answer: A

75. Which structure describes the high-level organization of a software system?
A. Software Architecture
B. Function Tree
C. Code Skeleton
D. Syntax Tree
Answer: A

76. What is the primary goal of Architectural Styles in Software Design?
A. Defining a set of constraints and rules for a system's structure
B. Ignoring non-functional requirements
C. Implementing UI elements
D. Handling memory management
Answer: A

77. Which pattern is used to create a family of related objects?
A. Abstract Factory Pattern
B. Composite Pattern
C. Strategy Pattern
D. Visitor Pattern
Answer: A

78. What is a major advantage of using Design Patterns in software construction?
A. Reducing code complexity and improving reusability
B. Ignoring dependencies
C. Increasing maintenance costs
D. Creating tightly coupled components
Answer: A

79. What is the main focus of Defensive Programming?
A. Writing code that anticipates potential errors
B. Ignoring error handling

C. Delaying code review
D. Prioritizing user interface
Answer: A

80. Which type of exception handling promotes robustness?
A. Graceful Degradation
B. Ignoring Exceptions
C. Failing Silently
D. Propagating All Errors
Answer: A

81. Which component ensures modularity in the system’s design?
A. Architectural Structures
B. Function Pointers
C. Syntax Trees
D. Global Variables
Answer: A

82. Which aspect of design involves specifying the responsibilities of different modules?
A. Architectural Viewpoints
B. State-Based Models
C. Event Handlers
D. Buffer Management
Answer: A

83. Which design principle ensures that a system is easily understandable and maintainable?
A. Simplicity
B. Obfuscation
C. Complication
D. Abstraction
Answer: A

84. What does ‘Design Patterns’ primarily provide?
A. Reusable solutions to common design problems
B. Debugging tools
C. Optimization techniques
D. Platform-specific implementation
Answer: A

85. Which strategy helps in designing Distributed Systems?
A. Implementing middleware for inter-service communication
B. Using single-threaded execution
C. Ignoring latency
D. Hardcoding configurations
Answer: A

86. Which style is useful in creating a system with a central control point?
A. Client-Server Architecture
B. Peer-to-Peer Network
C. Layered Architecture
D. Pipeline Architecture

Answer: A

87. What is the goal of Families of Programs?
    A. Sharing common functionality between related software systems
    B. Creating isolated systems
    C. Implementing monolithic architectures
    D. Using redundant code

    Answer: A

88. Which design principle ensures that objects in a program are replaceable with instances of their subtypes without altering the correctness of the program?
    A. Liskov Substitution Principle
    B. Interface Segregation Principle
    C. Open-Closed Principle
    D. Dependency Inversion Principle

    Answer: A

89. Which concept in software design helps separate the responsibilities of user interface, business logic, and data handling?
    A. Model-View-Controller (MVC)
    B. Single Responsibility Principle
    C. Factory Pattern
    D. Observer Pattern

    Answer: A

90. What does the term "Loose Coupling" refer to in software design?
    A. Dependencies between components are minimized
    B. Classes share a high number of attributes
    C. Modules are directly dependent on each other
    D. Functions use global variables extensively

    Answer: A

91. What is the main advantage of using Generics in software design?
    A. Code reusability and type safety
    B. Dynamic memory allocation
    C. Enhanced error handling
    D. Lower performance overhead

    Answer: A

92. Which design pattern is used to encapsulate a request as an object, thereby allowing for parameterization of clients with different requests?
    A. Command Pattern
    B. Decorator Pattern
    C. Prototype Pattern
    D. Composite Pattern

    Answer: A

93. What is the purpose of using a Builder Pattern in software design?
    A. To construct a complex object step-by-step
    B. To create a single object instance
    C. To monitor changes in objects

D. To share states between objects

Answer: A

94. Which type of testing is typically performed to evaluate a software system’s usability and user interface?

A. Usability Testing
B. Unit Testing
C. Integration Testing
D. Regression Testing

Answer: A

95. What is the primary role of an Interface Segregation Principle?

A. To ensure that no client is forced to implement unnecessary methods
B. To reduce system performance
C. To introduce global access
D. To combine multiple unrelated interfaces

Answer: A

96. What is the advantage of using State-Based Construction Techniques?

A. Simplifies complex state management
B. Increases code redundancy
C. Complicates event handling
D. Reduces program modularity

Answer: A

97. Which type of design pattern is used to provide a way to access elements of an aggregate object sequentially?

A. Iterator Pattern
B. Singleton Pattern
C. Flyweight Pattern
D. Adapter Pattern

Answer: A

98. Which of the following principles recommends separating the abstraction of a module from its implementation?

A. Dependency Inversion Principle
B. Adapter Pattern
C. Bridge Pattern
D. Template Method

Answer: C

99. Which component is typically used for data persistence in software systems?

A. Database
B. Middleware
C. Client Interface
D. Event Handler

Answer: A

100. Which approach is used for handling multiple, simultaneous processes within a software application?

A. Concurrency

B. Serial Execution
C. Lazy Loading
D. Deferred Execution

Answer: A