Which of the following is NOT a key principle of software design?

1. Abstraction
2. Modularity
3. Redundancy
4. Cohesion 3

What is the primary purpose of modularity in software design?

1. To increase the system’s complexity
2. To divide a system into smaller, manageable components
3. To duplicate code for better performance
4. To avoid encapsulation 2

Which principle of Object-Oriented Design promotes code reusability?

1. Encapsulation
2. Polymorphism
3. Inheritance
4. Cohesion 3

What does an interface define in a system?

1. The hardware of a system
2. The interaction between system components
3. The number of users in a system
4. The programming languages used 2

What does the architecture of a system define?

1. The programming language used
2. The structure, behavior, and views of the system
3. The cost of development
4. The number of users 2

Why is system architecture important?

1. It provides a high-level blueprint of how the system is structured
2. It ensures that all code is written in the same language
3. It guarantees system security
4. It eliminates the need for testing 1

The \_\_\_\_\_\_\_\_\_\_\_\_\_ Identifies software work as a system.

1. Detailed design
2. High level design
3. Architectural design
4. None 3

What does coupling refer to in software design?

1. The strength of a module’s internal relationships
2. The degree of interdependence between modules
3. The number of functions within a module
4. The amount of data stored in a module 2

What is the ideal combination for good software design?

1. High cohesion and high coupling
2. Low cohesion and low coupling
3. Low cohesion and high coupling
4. High cohesion and low coupling 4

If a module passes a flag variable to another module to control its behavior, this is an example of:

1. Control Coupling
2. Data Coupling
3. Content Coupling
4. Procedural Coupling 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_can be called as the global coupling.

1. Data coupling
2. Common coupling
3. Content coupling
4. Stamp coupling 2

What is the most appropriate word to describe the significance of software design in the SDLC?

1. Accuracy
2. Efficiency
3. Quality
4. Complexity 3

Which of the following is not a common characteristic of all design methods?

1. Functional component representation
2. Configuration management
3. Quality assessment guidelines
4. Refinement heuristics 2

What type of cohesion is being exhibited, If all tasks must be executed in the same time-span

1. Functional Cohesion
2. Temporal Cohesion
3. Functional Cohesion
4. Sequential Cohesion 2

What is the key characteristic of a well-designed data model?

1. High complexity and redundancy
2. Low maintainability
3. Efficient organization and minimal redundancy
4. No relationships between data elements 3

What is the most desirable form of coupling in software design?

1. Common Coupling
2. Stamp Coupling
3. Data Coupling
4. Content Coupling 3

A team is developing an API for an e-commerce platform. They notice that the interface between the payment gateway and the order processing system is poorly defined, leading to failed transactions and incorrect order statuses. Which step should the team take to fix this issue?

1. Clearly define interface specifications and how data will be exchanged
2. Remove API dependencies and handle payments manually
3. Allow all modules to modify each other’s data directly
4. Ignore interface design and focus only on back-end logic 1

Coupling and cohesion metrics are the metrics in which of the following design level

1. Architectural design
2. User interface design
3. Component level design
4. Pattern based design 3

A software development team is working on a customer support system. They find that the ticket management module is tightly coupled with the notification module, meaning any change in ticket management logic affects the notification system. What should the team do to improve the design?.

1. Reduce coupling by introducing an intermediary service for notifications
2. Merge both modules into one class for simplicity
3. Make the ticket management module dependent on multiple modules
4. Keep the design as it is because tight coupling is beneficial 1

Which of the following represents the worst type of coupling in software design, leading to the highest level of dependency between modules?

1. Content Coupling
2. Control Coupling
3. Stamp Coupling
4. Data Coupling 1

Which of the following is NOT a valid component of a class diagram?

1. Attributes
2. Operations
3. Use cases
4. Associations 3

In a class diagram, which relationship represents a “whole-part” association?

1. Aggregation
2. Inheritance
3. Dependency
4. Association 1

Which type of relationship is depicted using a solid line with a hollow triangle in a class diagram?

1. Generalization
2. Aggregation
3. Composition
4. Association 1

In a Banking System class diagram, which of the following is a suitable relationship?

1. Customer ⟶ has-a ⟶ Account
2. Bank ⟶ owns ⟶ Customer
3. Transaction ⟶ inherits ⟶ Customer
4. Account ⟶ manages ⟶ Bank 1

Which of the following is NOT a level of software testing?

1. Unit Testing
2. Integration Testing
3. Structural Testing
4. System Testing 3

What is the primary focus of unit testing?

1. Testing the entire system
2. Testing individual components or modules
3. Testing interactions between modules
4. Testing real-world scenarios 2

What is the primary purpose of a class diagram in UML?

1. To show interactions between objects over time
2. To describe the static structure of a system
3. To represent the sequence of events in a process
4. To define user interactions with the system 2

If a Department can exist without a College, but a College consists of multiple Departments, which relationship should be used?

1. Generalization
2. Aggregation
3. Composition
4. Association 2

Which testing level ensures that different modules or components work together correctly?

1. Unit Testing
2. System Testing
3. Integration Testing
4. Regression Testing 3

Which of the following is NOT a benefit of unit testing?

1. Helps identify bugs early in development
2. Improves code quality and maintainability
3. Replaces the need for integration testing
4. Facilitates code refactoring with confidence 3

What is the best time to perform unit testing in the software development lifecycle (SDLC)?

1. After integration testing
2. After system testing
3. During the development phase
4. At the end of the project 3

In Test-Driven Development (TDD), when are unit tests written?

1. After coding is completed
2. Before writing the actual code
3. During system testing
4. Only after user acceptance testing 2

What kind of errors does unit testing help to detect early?

1. Integration errors
2. System-wide performance issues
3. Logic errors in individual functions or modules
4. Security vulnerabilities 3

A software development team is designing a class diagram for a Car Rental System. They need to represent the relationship between a Car and an Engine, where an engine cannot exist independently of a car. Which type of relationship should they use?

1. Association
2. Aggregation
3. Composition
4. Generalization 3

In a Library Management System, a librarian needs to keep track of books borrowed by members. Each member can borrow multiple books, but a book can only be borrowed by one member at a time. Which class diagram relationship best represents this scenario?

1. One-to-One
2. One-to-Many
3. Many-to-Many
4. Aggregation 2

A developer is working on a Login Module that verifies usernames and passwords. He wants to test individual functions like validateUsername() and checkPassword() before integrating them with the database. Which type of testing is most appropriate?

1. Unit Testing
2. System Testing
3. Integration Testing
4. Acceptance Testing 1

A Hospital Management System needs to represent the relationship between Doctors, Patients, and Appointments. A doctor can have multiple appointments with different patients, and a patient can have multiple appointments with different doctors. How should this be represented in a class diagram?

1. Direct many-to-many association between Doctor and Patient
2. Introduce an Appointment class associating Doctor and Patient
3. Use inheritance between Doctor and Patient
4. Aggregation relationship between Doctor and Patient 2

In a Library Management System, each Book can have multiple Authors, and each Author can write multiple Books. Additionally, each Book has a unique ISBN and can have multiple Copies, each identified by a unique CopyID. How should these relationships be represented in a class diagram?

1. Book and Author have a one-to-many association; Book and Copy have a one-to-many association.
2. Book and Author have a many-to-many association; Book and Copy have a one-to-many aggregation.
3. Book and Author have a many-to-many association; Book and Copy have a one-to-many composition.
4. Book and Author have a one-to-one association; Book and Copy have a one-to-one composition. 3

In a complex Employee Management System, there are multiple types of employees such as FullTimeEmployee, PartTimeEmployee, and ContractEmployee. Each type has unique attributes and methods, but they also share common characteristics from a general Employee class. How should these relationships be represented in a class diagram?

1. Aggregation
2. Composition
3. Inheritance
4. Association 3

In an Online Shopping System, a Customer can place multiple Orders, and each Order can include multiple Products. Additionally, each Product can be part of multiple Orders. How should these relationships be represented in a class diagram?

* 1. Customer and Order have a one-to-many association; Order and Product have a many-to-many association.
  2. Customer and Order have a many-to-many association; Order and Product have a one-to-many association.
  3. Customer and Order have a one-to-one association; Order and Product have a many-to-many association.
  4. Customer and Order have a one-to-many association; Order and Product have a one-to-one association. 1

What is the main feature of the Waterfall Model in software development?

1. It allows for continuous development and iterations
2. It is a linear and sequential model
3. It uses multiple feedback loops for improvements
4. It is focused on Agile principles 2

Which of the following best describes an evolutionary process model?

1. The requirements are fully defined at the start
2. The product is built in small, iterative steps with ongoing feedback
3. The project is developed as a single complete version
4. The model focuses on strict, defined phases 2

In the Incremental Model, how software is developed?

1. All at once, after all requirements are gathered
2. In small, manageable parts with frequent releases
3. Only after final approval from clients
4. Without gathering requirements in advance 2

The Spiral Model is a combination of which two models?

1. Waterfall and Agile
2. Incremental and Waterfall
3. Waterfall and Evolutionary
4. Prototyping and Iterative 4

What is the duration of a typical sprint in a Scrum process?

1. 2-4 weeks
2. 1-2 days
3. 6 months
4. 1 year 1

What does agility refer to in context to software development?

1. Rapid project execution without any planning
2. Detailed documentation and strict timelines
3. Flexibility and adaptability to changes throughout the development process
4. Limited involvement of customers during development 3

What is the focus of the development process in Extreme Programming (XP)?

1. High-level documentation and planning
2. Fixed development phases
3. Customer collaboration and quick releases
4. Avoiding changes in requirements 3

Which of the following is a disadvantage of the Waterfall Model in software development?

1. It lacks a structured approach
2. It does not allow for changes during the development process
3. It is too flexible for large-scale projects
4. It encourages constant customer feedback throughout the development process 2

In the Incremental Model, which of the following is true about the delivery of the software product?

1. The product is delivered in segments, where each segment adds more functionality
2. The product is delivered as a single monolithic release after the entire development process
3. The product is delivered without completing any functionality, with the focus on planning
4. The product is released only after all requirements are fully defined 1

Which of the following is NOT one of the Agile Manifesto principles?

1. Customer collaboration over contract negotiation
2. Responding to change over following a plan
3. Processes and tools over individuals and interactions
4. Working software over comprehensive documentation 3

Which of the following practices is emphasized in Extreme Programming (XP)?

1. Comprehensive documentation before development starts
2. Extensive upfront design before coding begins
3. Fixed roles for developers with no flexibility
4. Pair programming, where two developers work on the same code 4

In Scrum, which of the following roles is responsible for defining the product backlog and ensuring it aligns with customer needs?

1. Product Owner
2. Scrum master
3. Development Team
4. Stakeholder 1

How does the Incremental Model handle changes to requirements during the development process?

1. Changes are impossible after the initial planning phase and are not considered.
2. Changes are incorporated after a complete version of the software is delivered.
3. Changes are implemented in successive increments, and each increment adds more functionality to the software.
4. Changes are considered, but only after the final release of the software.

3

Which metric is used to measure the complexity of a program during Basis Path Testing?

1. Test Coverage
2. Cyclomatic Complexity
3. Condition Testing
4. Path Length 2

In Loop Testing, which type of loop is the easiest to test?

1. Nested Loop
2. Simple Loop
3. Concatenated Loop
4. Recursive Loop 2

What is a key characteristic of Condition Testing?

1. It only verifies input/output interactions.
2. It examines individual Boolean conditions in decision statements.
3. It focuses on functional requirements.
4. It tests loops with high iterations. 2

Which of the following techniques is NOT part of White-Box Testing?

1. Basis Path Testing
2. Loop Testing
3. Boundary Value Analysis
4. Condition Testing 3

Which formula is used to calculate Cyclomatic Complexity?

1. V(G) = E – N + 2P
2. V(G) = N – E + 2P
3. V(G) = N + E – 2P
4. V(G) = E + N – 2P 1

In Loop Testing, what is the primary focus when testing nested loops?

1. To test the inner loop only once
2. To check loop boundaries independently
3. To analyze all possible iterations of all loops
4. To verify loop termination conditions 3

Which of the following is a valid test case for condition coverage?

1. Executing one condition of a decision
2. Evaluating all combinations of conditions
3. Running the code without any conditions
4. Checking only the “True” outcome 2

Which white-box testing technique requires identifying the linearly independent paths in the code?

1. Data Flow Testing
2. Basis Path Testing
3. Mutation Testing
4. Error Guessing 2

Why is white box testing often referred to as ""glass box"" testing?

1. Because it involves testing transparent software
2. Because the code is visible like glass
3. Because it only tests the user interface
4. Because it is fragile like glass 2

Which of the following correctly defines the independent paths in Basis Path Testing?

1. The paths that include only conditional statements
2. The paths that traverse each unique decision point at least once
3. The paths that execute all possible input values
4. The paths that contain only loop structures 2

What is the best way to test a nested loop using White-Box Testing?

1. Test only the outer loop with boundary conditions
2. Execute the inner loop once and the outer loop twice
3. Perform zero iterations, one iteration, and maximum iterations for both loops
4. Run the program with a single test case 3

Which statement about Cyclomatic Complexity is true?

1. A higher complexity always means a program has more bugs
2. It helps determine the number of independent test cases needed
3. It is calculated using the number of function calls
4. It is a measure of system performance 2

Which of the following test cases is essential to check the loop termination condition?

1. One test case where the loop runs zero times
2. One test case where the loop runs at least once
3. One test case where the loop executes the maximum number of iterations
4. All of the above 4

Which type of loop is the most complex to test in White-Box Testing?

1. Simple loop
2. Nested loop
3. Concatenated loop
4. Unstructured loop 2

Scenario: You are testing a payment processing system. The system applies a discount only if the user is a premium member and the purchase is above $100. Which conditions must you test for full decision coverage?

1. (premium=True, amount=120), (premium=False, amount=90)
2. (premium=True, amount=120), (premium=True, amount=80), (premium=False, amount=120)
3. (premium=True, amount=100), (premium=False, amount=100)
4. (premium=False, amount=80), (premium=True, amount=90) 2

"Scenario:

A loop in your code processes incoming sensor data. It terminates if it receives an error or after 100 iterations.What is the best approach to test this loop?"

1. Execute the loop once with no error
2. Inject an error on the 50th iteration and test boundary conditions (0, 100)
3. Run the loop to 100 iterations without errors
4. Check only the first and last iterations 2

"Scenario: Consider a function that calculates taxes based on age:

If age < 18: No tax

If 18 ≤ age ≤ 60: 20% tax

If age > 60: 10% tax

What test cases achieve full branch and condition coverage?"

1. (age=17), (age=25), (age=61)
2. (age=18), (age=59), (age=60)
3. (age=16), (age=30), (age=62)
4. (age=17), (age=18), (age=60), (age=61) 4

"A smart meter records energy consumption. A loop runs daily to verify the readings:

If consumption > threshold, alert is triggered.

If consumption < 0, an error is logged.

Which test cases ensure comprehensive loop and decision testing?"

1. (consumption=50), (consumption=150)
2. (consumption=0), (consumption=-1), (consumption=100)
3. (consumption=-5), (consumption=200)
4. (consumption=-1), (consumption=0), (consumption=150) 4

"Scenario:

A warehouse inventory system tracks product stock levels. The system follows these rules:

If stock > 100, mark the product as ""Overstocked.""

If 1 ≤ stock ≤ 100, mark it as ""In Stock.""

If stock == 0, mark it as ""Out of Stock.""

Negative stock values trigger an ""Error.""

Which test cases are required to achieve full branch and condition coverage?"

1. (stock=0), (stock=50), (stock=101)
2. (stock=-1), (stock=0), (stock=100), (stock=150)
3. (stock=-5), (stock=0), (stock=1), (stock=100), (stock=101)
4. (stock=0), (stock=1), (stock=50), (stock=101) 3