What is a "service-oriented architecture" (SOA)?
A) A way to build monolithic applications
B) An architectural pattern that relies on services to support business processes
C) A method to design graphical user interfaces
D) None of the above
Answer: B

In software architecture, "separation of concerns" refers to:
A) Mixing multiple functionalities
B) Dividing a system into distinct features that overlap minimally
C) Everything being tightly coupled
D) None of the above
Answer: B

Which of the following is a component of a layered architecture?
A) User Interface Layer
B) Code Layer
C) Hardware Layer
D) Network Layer
Answer: A
What is the main benefit of a layered architecture?

A) Increased coupling
B) Improved maintainability and separation of concerns
C) Easier to understand hardware requirements
D) Faster performance
Answer: B

What does the term "middleware" refer to in software architecture?
A) An application software layer closest to hardware
B) Software layers that connect two or more separate applications
C) The user interface of a software application
D) The final production environment
Answer: B

In a microservices architecture, services are typically:
A) Interdependent and tightly coupled
B) Monolithic and unified
C) Independently deployable and scalable
D) Hierarchical and rigid
Answer: C

What is a drawback of a monolithic architecture?
A) Simplified deployment
B) Easier debugging
C) Limited scalability and flexibility
D) Clearer code organization

Answer: C

Which of the following architectural styles focuses on processing and managing large volumes of data?
A) Event-Driven Architecture
B) RESTful Architecture
C) Big Data Architecture
D) Layered Architecture
Answer: C

What is the primary goal of object-oriented analysis?
A) To define the system's interface
B) To understand and document the problem domain
C) To build the actual system
D) To design the user interface
Answer: B

In OOAD, what is an "object"?
A) A procedural code block
B) An instance of a class that contains state (attributes) and behavior (methods)
C) A set of functions
D) A database record
Answer: B

Which of the following is NOT a principle of object-oriented programming?
A) Encapsulation
B) Inheritance
C) Polymorphism
D) Structured programming
Answer: D

What does the term "inheritance" refer to in OOAD?
A) Objects contain other objects
B) A new class derives attributes and methods from an existing class
C) Objects destroying old methods
D) None of the above
Answer: B

What is a "use case" in the context of OOAD?
A) A scenario describing a system's behavior as it responds to a request
B) A chart showing class relationships
C) A specific object instance
D) A method within a class
Answer: A

Which diagram is commonly used to capture use cases?
A) Class diagram
B) Sequence diagram
C) Use case diagram
D) Component diagram
Answer: C

What does "polymorphism" allow in object-oriented systems?
A) Multiple forms of an object or method
B) Strict type-checking of objects
C) Only one behavior per object
D) None of the above
Answer: A

What are "attributes" in an object-oriented class?
A) The methods defined in the class
B) The properties or characteristics of an object
C) The interfaces implemented by the class
D) The comments in the code
Answer: B

Which of the following best describes "composition" in OOAD?
A) One class inherits the properties of another
B) A class uses one or more objects of other classes in its methods
C) A method calls itself
D) None of the above
Answer: B

What is the main purpose of sequence diagrams in OOAD?
A) To define the structure of classes
B) To show the interactions between objects in a time sequence
C) To capture software requirements
D) To illustrate system architecture
Answer: B

What does API stand for?
A) Application Programming Interface
B) Application Program Interaction
C) Advanced Programming Integration
D) Application Protocol Interface
Answer: A

Which of the following is a common method of design verification?
A) Code walkthrough
B) User acceptance testing
C) Quality assurance
D) Performance testing
Answer: A

What does the acronym "UML" stand for in software design?
A) Unified Modeling Language
B) Universal Model Language
C) Unilateral Modeling Language
D) Unified Markup Language
Answer: A

Which of the following is NOT a design technique?
A) Wireframing

B) Prototyping
C) Performance Testing
D) Object-Oriented Analysis
Answer: C

What is the primary function of software architecture?
A) To create user interfaces
B) To define the system’s structure and behavior
C) To write code
D) To conduct user training
Answer: B

Which architecture style is useful for systems needing high scalability?
A) Monolithic
B) Layered
C) Microservices
D) Client-server
Answer: C

What is a "service-oriented architecture" (SOA)?
A) A way to build monolithic applications
B) An architectural pattern that relies on services to support business processes
C) A method to design graphical user interfaces
D) None of the above
Answer: B

Which of the following is an advantage of design for reuse?
A) Higher initial costs
B) Decreased productivity
C) Reduced development time for future projects
D) Increased system complexity
Answer: C

What is a "library" in the context of design for reuse?
A) A printed collection of code
B) A shared collection of precompiled routines that programs can use
C) A database of user interfaces
D) A block of code written without documentation
Answer: B

Which of the following is a design pattern specifically aimed at promoting reuse?
A) Singleton
B) Observer
C) Factory Method
D) All of the above
Answer: D

What is “refactoring” in software design?

A) Rewriting the code from scratch
B) Changing code without modifying its external behavior to improve its structure and maintainability
C) Combining multiple classes into one
D) None of the above
Answer: B

Which technique is used to identify reusable components?
A) Code obfuscation
B) Component analysis
C) Closed-box testing
D) Complete refactoring
Answer: B

Which of the following best describes "component-based design"?
A) A focus on monolithic applications
B) Building software applications from reusable components
C) Only involving libraries
D) Completing projects without user feedback
Answer: B

What is the purpose of a common interface when designing for reuse?
A) To increase coupling between components
B) To restrict component visibility
C) To allow different implementations to be interchangeable
D) To provide complex implementations
Answer: C

Which of the following is a challenge in design for reuse?
A) Increased resource consumption
B) Reduced marketability
C) Difficulty in testing reused components
D) Simplified design processes
Answer: C

What is a design pattern?
A) A one-time solution to a specific problem
B) A proven, reusable solution to a recurring design problem in software design
C) A template for managing software releases
D) A framework for coding
Answer: B

Which of the following is an example of a creational design pattern?
A) Observer
B) Singleton
C) Strategy
D) Composite
Answer: B

What does the Factory Method pattern do?
A) Creates objects but allows subclasses to alter the type of objects that will be created
B) Ensures a class has only one instance

C) Provides a way to use multiple algorithms interchangeably
D) Combines elements into a new structure
Answer: A

Which design pattern encourages communication between loosely coupled objects?
A) MVC (Model-View-Controller)
B) Adapter
C) Observer
D) Prototype
Answer: C

In which context would you use the Strategy design pattern?
A) When you want to represent a part-whole relationship
B) When you have multiple algorithms for a task and want to switch between them
C) When you need to construct objects with complex configurations
D) When you want to ensure the class has only one instance
Answer: B

What is the intent of the Adapter pattern?
A) To allow incompatible interfaces to work together
B) To create an interface that is easy to use
C) To ensure a single instance of a class
D) To remove the conditions of inheritance
Answer: A

What role does the Composite pattern serve?
A) To create families of related objects
B) To treat individual objects and compositions uniformly
C) To provide an interface for creating families of related objects
D) To separate algorithm implementation from its usage
Answer: B

The Decorator pattern is mainly used for:
A) Creating objects with common interface
B) Adding new responsibilities to objects dynamically
C) Ensuring a class has only one instance
D) Composing objects into tree structures
Answer: B

Which of the following is a behavioral design pattern?
A) Singleton
B) Builder
C) Mediator
D) Factory
Answer: C

What does the Visitor pattern allow you to do?
A) Add new operations to classes without modifying them

B) Define a family of algorithms
C) Create objects without specifying their concrete classes
D) Keep track of the changes in state
Answer: A

What is a class browser?
A) A tool that allows navigation through code or software components
B) A web browser to access online tutorials
C) A database tool for storing application data
D) A program used to generate reports
Answer: A

What is the purpose of a common interface when designing for reuse?
A) To increase coupling between components
B) To restrict component visibility
C) To allow different implementations to be interchangeable
D) To provide complex implementations
Answer: C

Which of the following is a challenge in design for reuse?
A) Increased resource consumption
B) Reduced marketability
C) Difficulty in testing reused components
D) Simplified design processes
Answer: C

What is a design pattern?
A) A one-time solution to a specific problem
B) A proven, reusable solution to a recurring design problem in software design
C) A template for managing software releases
D) A framework for coding
Answer: B

Which of the following is an example of a creational design pattern?
A) Observer
B) Singleton
C) Strategy
D) Composite
Answer: B

What does the Factory Method pattern do?
A) Creates objects but allows subclasses to alter the type of objects that will be created
B) Ensures a class has only one instance
C) Provides a way to use multiple algorithms interchangeably
D) Combines elements into a new structure
Answer: A

Which design pattern encourages communication between loosely coupled objects?
A) MVC (Model-View-Controller)
B) Adapter
C) Observer
D) Prototype

Answer: C

In which context would you use the Strategy design pattern?
A) When you want to represent a part-whole relationship
B) When you have multiple algorithms for a task and want to switch between them
C) When you need to construct objects with complex configurations
D) When you want to ensure the class has only one instance
Answer: B

What is the intent of the Adapter pattern?
A) To allow incompatible interfaces to work together
B) To create an interface that is easy to use
C) To ensure a single instance of a class
D) To remove the conditions of inheritance
Answer: A

What role does the Composite pattern serve?
A) To create families of related objects
B) To treat individual objects and compositions uniformly
C) To provide an interface for creating families of related objects
D) To separate algorithm implementation from its usage
Answer: B

The Decorator pattern is mainly used for:
A) Creating objects with common interface
B) Adding new responsibilities to objects dynamically
C) Ensuring a class has only one instance
D) Composing objects into tree structures
Answer: B

Which of the following is a behavioral design pattern?
A) Singleton
B) Builder
C) Mediator
D) Factory
Answer: C

What does the Visitor pattern allow you to do?
A) Add new operations to classes without modifying them
B) Define a family of algorithms
C) Create objects without specifying their concrete classes
D) Keep track of the changes in state
Answer: A

What is a class browser?
A) A tool that allows navigation through code or software components
B) A web browser to access online tutorials
C) A database tool for storing application data

D) A program used to generate reports
Answer: A

Which of the following does a class browser typically provide?
A) Execution of code
B) Visualization of class relationships and hierarchy
C) Code optimization features
D) Data storage options
Answer: B

What is the purpose of a component in component-based computing?
A) To serve as a single point in a monolithic system
B) To encapsulate behavior and data, providing a clear interfaces
C) To define user interface elements
D) To maximize code obfuscation
Answer: B

Which of the following is a benefit of component-based computing?
A) Increased development time
B) Enhanced code reuse
C) Limited scalability
D) Complex integration requirements
Answer: B

What is meant by "plug-and-play" in the context of components?
A) Components can be interchanged with minimal effort
B) Components are hardwired into the system
C) All components require a restart to work together
D) None of the above
Answer: A

Which technique allows different components to communicate in component-based systems?
A) Inheritance
B) Function Overriding
C) Interfaces/Contracts
D) Static Typing
Answer: C

What is the role of a component container?
A) To store components on disk
B) To manage the lifecycle of components and facilitate their interactions
C) To provide user interfaces
D) To compile code into executable forms
Answer: B

What does "binding" refer to in component-based computing?
A) Connecting components at compile time
B) Integrating different programming languages
C) Linking a component to its external interfaces
D) Compressing component files for storage
Answer: C

Which of the following is a challenge of component-based computing?
A) Enhanced interoperability
B) Component versioning and compatibility
C) Simplified integration
D) Independent deployment
Answer: B

In a class browser, what does "refactoring" mean?
A) Compiling source code into machine code
B) Reorganizing existing code without changing its behavior
C) Deleting all unnecessary classes
D) Adding comments to code
Answer: B

What is requirements analysis?
A) The process of coding the system
B) The practice of documenting user needs and expectations
C) Budgeting for software development
D) Designing the database schema
Answer: B

Which of the following are requirements types?
A) Functional and non-functional
B) Code and design
C) Static and dynamic
D) All of the above
Answer: A

What is a functional requirement?
A) A requirement related to system performance
B) A requirement specifying what the system should do
C) A non-essential requirement
D) A requirement about system usability only
Answer: B

Which of the following is a non-functional requirement?
A) The system shall allow users to log in
B) The system shall have an availability of 99.9%
C) The system shall process transactions
D) The system must store user data securely
Answer: B

What is the purpose of a requirements specification document?
A) To outline the architectural decisions
B) To detail the requirements that the software must fulfill
C) To define the project's budget
D) To provide guidelines for testing

Answer: B

Who is primarily responsible for gathering requirements in software development?
A) Developers
B) Project Managers
C) Business Analysts
D) Quality Assurance Engineers
Answer: C

What is a common technique used to elicit requirements from stakeholders?
A) Code review
B) Prototyping
C) Walkthroughs
D) Test case creation
Answer: B

What does “stakeholder” mean in the context of requirements analysis?
A) Individuals who write the code
B) Anyone impacted by or invested in the project outcome
C) Validators of requirements
D) Qualitative researchers
Answer: B

Which of the following activities is NOT part of requirement analysis?
A) Eliciting requirements from stakeholders
B) Verifying system performance
C) Documenting requirements
D) Prioritizing requirements
Answer: B

What is the role of use cases in requirements analysis?
A) To represent external system behavior and interactions
B) To define the design architecture
C) To validate the final product
D) To perform risk analysis
Answer: A

Which of the following is a benefit of component-based design?
A) Enhanced system rigidity
B) Easier system modifications
C) Increased development costs
D) Longer testing periods
Answer: B

In design terminology, which of the following refers to an "interface"?
A) A class that implements behavior
B) A contract that defines methods or properties without implementation
C) A visualization of the software
D) A storage medium for software
Answer: B

What is a "design review"?

A) An assessment of code efficiency
B) A meeting to evaluate a design for flaws or improvements before implementation
C) A performance evaluation for software
D) An agreement on project objectives
Answer: B