# Chapter 1: Introduction

### What is software modeling?

* 1. Developing models of software. (b) Designing software applications before coding.

(c) Developing software diagrams. (d) Developing software prototypes

### What is the Unified Modeling Language?

* 1. A programming language for describing object-oriented models.
  2. A diagramming tool for drawing object-oriented models.
  3. A graphical language for describing object-oriented models.
  4. A standardized graphical language and notation for describing object oriented models.

### What is a software architecture?

* 1. The software inside a building. (b) The structure of a client/server system.

(c) The overall structure of a software system. (d) The software classes and their relationships.

### What is a software design notation?

* 1. Notes about the software design. (b) A graphical or textual description of the software.

(c) Documentation of the software. (d) A systematic approach for producing a design.

### What is a software design concept?

* 1. A graphical or textual description of the software. (b) Documentation of the software.

1. A fundamental idea that can be applied to designing a system.
2. A systematic approach for producing a design.

### What is a software design strategy?

* 1. A graphical or textual description of the software. (b) A fundamental idea that can be applied to designing a system.

(c) A systematic approach for producing a design. (d) An overall plan and direction for developing a design.

### What are software structuring criteria?

* 1. Fundamental ideas that can be applied to designing a system.
  2. Systematic approaches for producing a design.
  3. Guidelines used to help in structuring a software system into its components.
  4. Overall plans for developing a design.

### What is a software design method?

* 1. A systematic approach for producing a design.
  2. Guidelines used to help in structuring a software system into its components.
  3. An overall plan for developing a design.
  4. A graphical or textual description of the software.

### What is a platform-independent model (PIM)?

* 1. A software platform before a commitment is made to a specific hardware platform.
  2. A precise model of the software architecture before a commitment is made to a specific platform.
  3. A precise model of the software architecture mapped to a specific platform.
  4. A graphical or textual description of the software.

### What is a platform-specific model (PSM)?

* 1. A specific hardware platform.
  2. A precise model of the software architecture before a commitment is made to a specific platform.
  3. A precise model of the software architecture mapped to a specific platform.
  4. A graphical or textual description of the software.

# Chapter 2: Overview of the UML Notation

### How is an actor depicted on a use case diagram?

(a) An oval (b) A stick figure (c) A box (d) A dashed line

### How is a use case depicted on a use case diagram?

(a) An oval (b) A stick figure (c) A box (d) A dashed line

### How is a class depicted on a class diagram?

* 1. A box with one compartment (b) A box with one or two compartments

(c) A box with one, two, or three compartments (d) An oval

### How is an association depicted on a class diagram?

* 1. A solid line joining two class boxes (b) A dashed line joining two class boxes

(c) A diamond touching the upper class box (d) An arrowhead touching the upper class box

### How is public visibility depicted for a class element on a class diagram?

(a) + sign (b) − sign (c) # sign (d) ∗sign

### What are the two kinds of UML interaction diagrams?

* 1. Class diagram and sequence diagram (b) Sequence diagram and communication diagram

(c) Class diagram and communication diagram (d) Statechart and communication diagram

### What does an interaction diagram depict?

* 1. Objects and links (b) Classes and relationships

(c) Objects and messages (d) States and events

### What does a statechart diagram depict?

* 1. Objects and links (b) Classes and relationships

(c) Objects and messages (d) States and events

### What is a UML package?

* 1. A box (b) A grouping of classes

(c) A grouping of use cases (d) A grouping of model elements

### What does a deployment diagram depict?

* 1. The physical configuration of the system in terms of physical classes and physical connections between the classes
  2. The physical configuration of the system in terms of physical objects and physical connections between the objects
  3. The physical configuration of the system in terms of physical nodes and physical connections between the nodes
  4. The physical configuration of the system in terms of physical computers and physical networks between the computers

# Chapter 3: Software Life Cycle

**Models and Processes**

### What is a software life cycle?

* 1. The life of the software (b) A cyclic approach to developing software

(c) A phased approach to developing software (d) The life of software developed in cycles

### What is the waterfall life cycle model?

* 1. Software developed under a waterfall (b) A process model in which each phase is completed before the next phase is started

1. A process model in which phases are overlapped
2. A process model in which phases are cyclic

### Which of the following is a limitation of the waterfall life cycle model?

* 1. Software is developed in phases. (b) Each phase is completed before the next phase is started.

(c) Software development is cyclic. (d) Software requirements are not properly tested until a working system is available.

### Which of the following approaches can overcome the limitation in the previous question?

* 1. Phased software development (b) Throwaway prototyping

(c) Evolutionary prototyping (d) Incremental development

### What is evolutionary prototyping?

* 1. Phased software development (b) Throwaway prototyping

(c) Risk-driven development (d) Incremental development

### What approach does the spiral model emphasize?

* 1. Phased software development (b) Throwaway prototyping

(c) Risk-driven development (d) Incremental development

### What is the goal of software validation?

* 1. Building the system (b) Building the right system

(c) Building the system right (d) Testing the system

### What is the goal of software verification?

* 1. Building the system (b) Building the right system

(c) Building the system right (d) Testing the system

### What is “white box” testing?

(a) Unit testing (b) Integration testing (c) Testing with knowledge of the system internals

(d) Testing without knowledge of the software internals

### What is “black box” testing?

(a) System testing (b) Integration testing (c) Testing with knowledge of the system internals

(d) Testing without knowledge of the software internals

# Chapter 4: Software Design and Architecture Concepts

### Which of the following are object oriented concepts?

(a) Modules and interfaces (b) Modules and information hiding

(c) Classes, information hiding, and inheritance (d) Concurrency and information hiding

### Which of the following is a characteristic of an object?

* 1. A function or subroutine (b) A module

1. Groups data and procedures that operate on the data
2. Groups a function and an algorithm

### What is a class?

* 1. An object instance (b) The implementation of the object

1. A collection of objects with the same characteristics
2. A collection of objects with different characteristics

### What is an operation (also known as method) of a class?

* 1. Specification and the implementation of a function performed by a class
  2. Specification and the implementation of a subroutine provided by a class
  3. Specification and the implementation of a function or procedure provided by a class
  4. Specification and the implementation of an interface provided by a class

### What is the signature of an operation?

* 1. The operation’s name
  2. The operation’s function or subroutine
  3. The operation’s name, parameters, and return value
  4. The object’s interface

### What is the interface of a class?

* 1. The signature of a class (b) The specification of operations provided by the class

(c) The internals of the class (d) The implementation of the class

### What is an attribute?

* 1. A description of a class (b) An internal property of a class

(c) A data item held by a class (d) A parameter of a class

### What is information hiding in software design?

* 1. Hiding information so that it cannot be found
  2. Hiding a design decision that is considered likely to change
  3. Hiding information to make it secure
  4. Encapsulating data in a class

### What is data abstraction?

* 1. Another name for information hiding
  2. Encapsulating data so that its structure is hidden
  3. Storing data in a database
  4. Storing data in a data structure

### What is inheritance?

* 1. A mechanism for inheriting characteristics from a parent
  2. A mechanism for sharing and reusing code between classes
  3. A mechanism for sharing data between classes
  4. A mechanism for hiding information between classes

# Chapter 5: Overview of

**Software Modeling and Design Method**

### What is carried out during requirements modeling?

* 1. Functional requirements of the system are described in terms of functions, inputs, and outputs.
  2. Functional requirements of the system are described in terms of actors and use cases.
  3. Functional requirements of the system are described textually.
  4. Functional requirements of the system are determined by interviewing users.

### What is carried out during analysis modeling?

* 1. Developing use case models
  2. Developing data flow and entity relationship diagrams
  3. Developing static and dynamic models
  4. Developing software architectures

### What is carried out during design modeling?

* 1. Developing use case models
  2. Developing data flow and entity relationship diagrams
  3. Developing static and dynamic models
  4. Developing software architectures

### What is carried out during incremental software construction?

* 1. Detailed design and coding of the classes in a subset of the system
  2. Detailed design, coding, and unit testing of the classes in a subset of the system
  3. Coding and unit testing of the classes in a subset of the system
  4. Unit and integration testing of the classes in a subset of the system

### What is carried out during incremental software integration?

* 1. Implementation of the classes in each software increment
  2. Unit testing of the classes in each software increment
  3. Integration testing of the classes in each software increment
  4. System testing of the classes in each software increment

### What is carried out during system testing?

(a) White box testing (b) Black box testing (c) Unit testing (d) Integration testing

# Chapter 6: Use Case Modeling

### What is a use case?

* 1. A case study involving users (b) A sequence of interactions between the user and the system

1. A sequence of interactions between the user and the objects in the system
2. A sequence of user inputs to the system

### What is an actor in a use case?

* 1. An object inside the system (b) A person who performs on stage

1. An external entity that interacts with the system
2. The customer to whom the system will be delivered

### What is a primary actor?

* 1. The actor who goes on stage first (b) The actor that starts the use case

(c) An actor that participates in the use case (d) An object inside the system

### What is a secondary actor?

* 1. The actor who goes on stage second (b) The actor that starts the use case

(c) An actor that participates in the use case (d) An object inside the system

### What is an alternative sequence in a use case?

* 1. A sequence that describes an error case
  2. A sequence that is different from the main sequence
  3. A sequence that describes interactions with a secondary actor
  4. A sequence that describes interactions with a primary actor

### What can an inclusion use case be used for?

* 1. To describe an inclusive use case (b) To describe a lengthy interaction with an actor

1. To describe functionality that is common to more than one use case
2. To describe a use case that includes other use cases

### What can an extension use case be used for?

* 1. To describe a lengthy interaction with an actor
  2. To describe functionality that is common to more than one use case
  3. To describe the functionality of a use case that is extended by another use case(s)
  4. To describe a conditional part of a different use case that is only executed under certain circumstances

### What can an activity diagram be used for in use case modeling?

* 1. To depict the sequence of activities executed by all the use cases in the system
  2. To depict the sequence of external activities that the use case interacts with
  3. To depict the sequence of active objects in a use case
  4. To depict the activities in the main and alternative sequences of a use case

### How can a nonfunctional requirement be described in a use case model?

* 1. In a separate section of the use case description
  2. As a use case precondition (c) As a use case postcondition

(d) In a separate document

### What is a use case package?

* 1. A package describing the actors in the system (b) A package describing the use cases in the system

(c) A group of related use cases (d) The package of objects that participate in the use case

# Chapter 7: Static Modeling

### What is a class?

(a) A course (b) An object instance

(c) A client or server in the system (d) A collection of objects with the same characteristics

### What is an attribute?

* 1. A relationship between two classes (b) A parameter of an operation or method

(c) A data value held by an object in a class (d) The return value from an operation

### What is an association?

* 1. A relationship between two classes (b) A relationship between two objects

(c) A link between two classes (d) A link between two objects

### What is meant by the multiplicity of an association?

* 1. The number of associations in a class
  2. The number of associations between two classes
  3. How many instances of one class relate to how many instances of another class
  4. How many instances of one class relate to a single instance of another class.

### What is an association class?

* 1. A class with multiple associations
  2. A class with one association
  3. A class that models an association between two or more classes
  4. A class that models an association between two or more objects

### What is a generalization/specialization hierarchy?

* 1. A whole/part relationship (b) An inheritance relationship

(c) An association between a generalized class and a specialized class (d) A layered hierarchy

### What is a composition hierarchy?

* 1. A weak form of a generalization/ specialization hierarchy
  2. A strong form of a generalization/ specialization hierarchy
  3. A weak form of a whole/part relationship (d) A strong form of a whole/part relationship

### What is an aggregation hierarchy?

* 1. A weak form of a generalization/ specialization hierarchy
  2. A strong form of a generalization/specialization hierarchy
  3. A weak form of a whole/part relationship (d) A strong form of a whole/part relationship

### What does the system context class diagram define?

* 1. The entity classes in the system (b) How the system interfaces to other systems

1. The boundary between the system and the external environment
2. The context classes in the system

### What is an entity class?

* 1. A class on an entity/relationship diagram (b) A class that stores data

(c) A class that interfaces to an external entity (d) An external class

## Chapter 8: Object and Class Structuring

### What is a boundary object?

* 1. An external object (b) An object that stores data

1. An object that communicates with an external object
2. An object that controls other objects

### What is a control object?

* 1. An object that depends on other objects (b) An object that communicates with an external object

(c) An object that controls other objects (d) An object that is controlled by other objects

### What is a state-dependent control object?

* 1. An object that depends on a state machine (b) An object that communicates with a state machine

(c) An object that controls a state machine (d) An object that executes a state machine

### What is a coordinator object?

* 1. A manager object (b) An object that makes decisions based on a state machine

(c) A decision-making object (d) An object that decides which entity object to interact with

### How would you determine a boundary class from the context diagram?

* 1. By looking at it (b) By selecting the external classes on the context diagram

1. By determining the software classes that communicate with the external classes
2. By drawing the boundary between the hardware and software classes

### What is a timer object?

* 1. An external clock (b) An internal clock

1. An object that is awakened by an external timer
2. An object that interacts with a clock

### What do class structuring criteria help with?

* 1. Structuring an application into classes (b) Defining the attributes of a class

(c) Defining the associations of a class (d) Defining the operations of a class

### What is the classification process for application classes analogous to?

* 1. Categorizing books in a library (b) Deciding how many copies of a book are needed

(c) Finding the classrooms in a school (d) Identifying what labs the school has

### What is the purpose of a stereotype in class structuring?

* 1. To label a class according to its class structuring criterion
  2. To identify the objects that belong to the same class
  3. To distinguish between external objects and software objects
  4. To identify the association between two classes

### What is a business logic object?

* 1. An object used in business applications
  2. An object that defines business specific application logic
  3. The internal logic of an object
  4. A business object that determines whether a client request is logical

# Chapter 9: Dynamic Interaction Modeling

### What does an interaction diagram depict?

* 1. The state and transitions inside a control object (b) Classes and their relationships

1. Software objects and the sequence of their interactions
2. The external objects communicating with the system

### How is an actor depicted on an interaction diagram?

* 1. An actor has an association with the interaction diagram.
  2. An actor can provide input to or receive output from a boundary object.
  3. An actor can provide input to or receive output from a boundary class.
  4. An instance of an actor can provide input to or receive output from a boundary object.

### What does a sequence diagram depict?

* 1. The sequence of external objects communicating with each other
  2. Classes and their relationships
  3. Software objects and the sequence of their interactions
  4. The external objects communicating with the system

### What does a communication diagram depict?

* 1. The sequence of external objects communicating with each other
  2. Classes and their relationships
  3. Software objects and the sequence of their interactions
  4. The external objects communicating with the system

### What is the instance form of an interaction diagram?

* 1. Depicts several object instances interacting with each other
  2. Depicts one possible sequence of interactions among object instances
  3. Depicts all possible interactions among object instances
  4. Depicts all object instances and their links to each other

### What is the generic form of an interaction diagram?

* 1. Depicts several objects interacting with each other
  2. Depicts one possible sequence of interactions among objects
  3. Depicts all possible interactions among objects
  4. Depicts all classes and their associations with each other

### During dynamic interaction modeling, use cases are realized as follows:

* 1. Determine objects that participate in each use case and the sequence of interactions among them.
  2. Determine external objects and the sequence in which they provide inputs to and receive outputs from each use case.
  3. Determine sequence of interactions among use cases.
  4. Determine how a use case is depicted through internal states and transitions between them.

### Which of the following interactions could happen on an interaction diagram?

* 1. An external user sends a message to a user interaction object.
  2. An external user sends a message to an entity object.
  3. An external user sends a message to an I/O object.
  4. An external user sends a message to a printer object.

### Which of the following interactions is NOT likely to happen on an interaction diagram?

* 1. A user interaction object sends a message to an entity object.
  2. An input object sends a message to a state-dependent control object.
  3. An input object sends a message to a printer object.
  4. A user interaction object sends a message to a proxy object.

### What kind of object would be the first object to receive an input from an external object?

(a) A user interaction object (b) A proxy object (c) An entity object (d) A boundary object

# Chapter 10: Finite State Machines

### What is a state in a state machine?

* 1. A recognizable situation that exists over an interval of time
  2. A condition that is True or False
  3. An input from the external environment (d) An output from the system

### What is an event in a state machine?

* 1. A discrete signal that causes a change of state
  2. An input from the external environment (c) An input that is True or False

(d) The result of a state transition

### What is an action in a state machine?

* 1. An occurrence at a point in time (b) A cause of a state transition

(c) An interval between two successive events (d) A computation that executes as a result of a state transition

### What is an entry action in a state machine?

* 1. An action that is performed when the state is entered
  2. An action that is performed when the state is left
  3. An action that starts executing when the state is entered and completes executing when the state is left
  4. An action that executes as a result of a state transition

### What is an exit action in a state machine?

* 1. An action that is performed when the state is entered
  2. An action that is performed when the state is left
  3. An action that starts executing when the state is entered and completes executing when the state is left
  4. An action that executes as a result of a state transition

### What is a condition used for in a state machine?

* 1. A conditional action (b) A conditional state

1. A conditional state transition
2. A conditional event

### What is a state transition into a composite state equivalent to?

* 1. A transition into only one of the substates (b) A transition into each of the substates

(c) A transition into none of the substates (d) A transition into any one of the substates

### What is a state transition out of a composite state equivalent to?

* 1. A transition out of only one of the substates (b) A transition out of each of the substates

1. A transition out of none of the substates
2. A transition out of any one of the substates

### How does a composite state relate to a substate?

* 1. A composite state is decomposed into substates.
  2. Composite states are composed into substates.
  3. A composite state transitions to a substate.
  4. A substate transitions to a composite state.

### If two actions are shown on a given state transition, which of the following is true?

* 1. The two actions are dependent on each other.
  2. The two actions are independent of each other.
  3. One action provides an input to the other action.
  4. The second action executes when the first action completes execution.

# Chapter 11: State-Dependent

**Dynamic Interaction Modeling**

## What does a state-dependent interaction involve?

(a) A control object (b) A state-dependent entity object

(c) A state-dependent control object (d) A state-dependent user interaction object

## Which kind of object executes a state machine

* 1. Any software object (b) An entity object

(c) A state-dependent control object (d) A statechart

## An input message to a state-dependent control object corresponds to:

* 1. An event on the internal state machine (b) An action on the internal state machine

(c) A condition on the internal state machine (d) A state on the internal state machine

## An output message from a state dependent control object corresponds to:

* 1. An event on the internal state machine (b) An action on the internal state machine

(c) A condition on the internal state machine (d) A state on the internal state machine

## An interaction diagram should be developed for:

* 1. Only the main sequence of the use case
  2. The main sequence and every alternative sequence of the use case
  3. The main sequence and a representative alternative sequence of the use case
  4. The alternative sequences of the use case

## Which of the following could happen on an interaction diagram?

* 1. A state-dependent control object sends a message to an entity object.
  2. A state-dependent control object sends a message to a coordinator object.
  3. A state-dependent control object sends a message to a printer object.
  4. All of the above

## If the same state machine is used in more than one use case, how is this modeled on interaction diagrams?

* 1. Develop one state-dependent control object for each use case.
  2. Develop one state-dependent control object containing states from each use case.
  3. Develop a hierarchical state machine. (d) Develop a coordinator object.

## How would two state-dependent control objects communicate with each other?

* 1. By sending messages to each other (b) By transitioning to the same state

(c) Through an entity object (d) Through a proxy object

## An object can send alternative messages a or b to a state-dependent control object. How is this handled in the state machine?

* 1. One state with a different transition out of it for each incoming message
  2. One state for each of the alternative messages
  3. A composite state to handle the alternative messages
  4. A substate for each alternative message

## In a system in which a client object executes a state machine and communicates with a service, which of the following is true?

* 1. The client has a state-dependent control object but the service does not.
  2. The service has a state-dependent control object but the client does not.
  3. Both the client and the service have state-dependent control objects.
  4. Neither the client nor the service has a state-dependent control object.

# Chapter 12: Overview of Software Architecture

## What does the software architecture describe?

* 1. The software inside a building (b) The structure of a client/server system

(c) The overall structure of a software system (d) The software classes and their relationships

## Which of the following statements is NOT true for a component?

* 1. A composite object composed of other objects (b) An operation

(c) A simple object (d) Provides an interface

## What is a structural view of a software architecture?

* 1. A view in terms of a module hierarchy (b) A view in terms of components and connectors

1. A view of the physical configuration in terms of nodes and interconnections
2. A view in terms of objects and messages

## What is a dynamic view of a software architecture?

* 1. A view in terms of a module hierarchy (b) A view in terms of components and connectors

1. A view of the physical configuration in terms of nodes and interconnections
2. A view in terms of objects and messages

## What is a deployment view of a software architecture?

* 1. A static view in terms of a module hierarchy
  2. A static view in terms of components and connectors
  3. A view of the physical configuration in terms of nodes and interconnections
  4. A dynamic interaction view in terms of objects and messages

## What is a software architectural pattern?

* 1. The structure of the major subsystems of a system
  2. The components and connectors in a software architecture
  3. A small group of collaborating objects
  4. A recurring architecture used in a variety of systems

## What happens in a Layers of Abstraction pattern?

* 1. Each layer uses services in the layer immediately below it.
  2. Each layer uses services in the layer immediately above it.
  3. Each layer uses services in the layers immediately above it and below it.
  4. Each layer is independent of the other layers.

## What happens in a Call/Return pattern?

* 1. A calling operation in the calling object sends a message to an operation (a.k.a. method) in the called object.
  2. A calling operation in the calling object invokes an operation (a.k.a. method) in the called object.
  3. The calling object waits for a response from the called object.
  4. The calling object does not wait for a response from the called object.

## A producer sends a message to a consumer. Which one of the following is asynchronous message communication?

* 1. The producer waits for a response from the consumer.
  2. The producer does not wait for a response from the consumer.
  3. The producer goes to sleep. (d) The producer waits for a timeout.

## A producer sends a message to a consumer. Which one of the following is synchronous message communication with reply?

* 1. The producer waits for a response from the consumer.
  2. The producer does not wait for a response from the consumer.
  3. The producer goes to sleep. (d) The producer waits for a timeout.

# Chapter 13: Software Subsystem Architectural Design

## What is an integrated communication diagram?

* 1. A communication diagram formed by combining objects
  2. A synthesis of all the communication diagrams developed to support the use cases
  3. A communication diagram depicting the objects that realize a use case
  4. A communication diagram that integrates the entity objects from the static model

## Which of the following objects should be assigned to the same subsystem?

* 1. Objects that are part of the same composite object (b) Client and server objects

(c) User interface and entity objects (d) Objects that are associated with each other

## Objects that are in geographically different locations should be:

* 1. In the same subsystem (b) In different subsystems

(c) In a composite subsystem (d) In layered subsystems

## If scope of control is used in subsystem structuring, then:

* 1. A user interface object is placed in the same subsystem as an entity object it updates.
  2. A state-dependent control object is placed in the same subsystem as the objects it controls.
  3. A state-dependent control object is placed in a different subsystem from the objects it controls.
  4. A user interface object is placed in a different subsystem from an entity object it updates.

## How should an external object be designed to interface to the system?

* 1. It should interface to one subsystem. (b) It should interface to several subsystems.

(c) It should interface to every subsystem. (d) It should interface to none of the subsystems.

## A user interface subsystem is a type of:

* 1. Control subsystem (b) Service subsystem

(c) Client subsystem (d) I/O subsystem

## Which of the following objects are NOT likely to be in the same subsystem?

* 1. User interface object and entity object
  2. State-dependent control object and coordinator object
  3. Business logic object and entity object (d) I/O object and state-dependent control object

## Which of the following subsystems is NOT likely to be a client subsystem?

* 1. Control subsystem (b) User interaction subsystem

(c) Service subsystem (d) I/O subsystem

## When is a coordinator subsystem required?

* 1. If the subsystem needs to coordinate several internal objects
  2. If the subsystem needs to coordinate multiple I/O devices
  3. If the subsystem receives messages from multiple client subsystems
  4. If the subsystem needs to coordinate the execution of other subsystems

## When is a control subsystem required?

* 1. If the subsystem needs to control several internal objects
  2. If the subsystem needs to control multiple I/O devices
  3. If the subsystem needs to control multiple client subsystems
  4. If the subsystem needs to control the execution of other subsystems

# Chapter 14: Designing Object- Oriented Software Architectures

## What is an information hiding object?

* 1. An active object that encapsulates data (b) A passive object that encapsulates data

(c) A class that encapsulates data (d) A task that encapsulates data

## What is a class interface?

* 1. Specifies the internals of the operations of a class
  2. Specifies the externally visible operations of a class
  3. Specifies the parameters of a class operation. (d) Specifies the signature of a class operation

## Which of the following is NOT an object-oriented concept?

* 1. Information hiding (b) Class

(c) Subclass (d) Subroutine

## Which of the following is a class that realizes an interface?

* 1. The class calls the interface. (b) The class implements the interface.

(c) The class is called by the interface. (d) The class is independent of the interface.

## Which of the following is an entity class?

* 1. An information hiding class (b) A subclass

(c) A control class (d) A data abstraction class

## What does a state machine class encapsulate?

* 1. A state transition table (b) A statechart

1. The current state of the machine
2. A state transition table and the current state of the machine

## Which of the following is unlikely to be a graphical user interface class?

* 1. A menu (b) A window

(c) A button (d) A pin

## Which of the following is unlikely to be encapsulated in a business logic class?

* 1. A business rule (b) Calls to operations of an entity class

(c) Deny cash withdrawal if balance of account is less than $10 (d) A dialog box

## Which of the following is NOT allowed through inheritance?

* 1. Subclass inherits attributes from superclass.
  2. Subclass inherits operations from superclass.
  3. Subclass redefines attributes inherited from superclass.
  4. Subclass redefines operations inherited from superclass.

## Which of the following is true for an abstract class?

* 1. It is used as a template for creating objects.
  2. It is used as a template for creating subclasses.
  3. It is used as a template for creating classes.
  4. It is used as a template for creating superclasses.

## In object-oriented design, polymorphism means that:

* 1. Different classes may have the same name.
  2. Different classes may have the same interface name.
  3. Different classes may have the same superclass name.
  4. Different classes may have the same operation name.

## With polymorphism and dynamic binding, an object can:

* 1. Invoke operation of different names on the same objects.
  2. Invoke operations of different names on different objects.
  3. Invoke an operation of the same name on the same object.
  4. Invoke an operation of the same name on different objects.

# Chapter 15: Designing

**Client/Server Software Architectures**

## What is a server?

* 1. A hardware/software system that serves customers
  2. A subsystem that makes requests and waits for the responses
  3. A subsystem that responds to requests from clients
  4. A hardware/software system that provides one or more services for multiple clients

## The basic client/single service architectural pattern states that:

* 1. Multiple clients request services, and multiple services fulfill client requests.
  2. Multiple clients request services, and a service fulfills client requests.
  3. A client requests services, and a service fulfills client requests.
  4. A client requests services, and multiple services fulfill client requests.

## In a Multi-tier Client/Service architectural pattern, which of the following is true about an intermediate tier?

* 1. An intermediate tier is a client tier. (b) An intermediate tier is a service tier.

1. An intermediate tier is both a control tier and a service tier.
2. An intermediate tier is both a client tier and a service tier.

## How is Multiple Client/Multiple Service architectural pattern different from a Multiple Client/Single Service architectural pattern?

* 1. A service can receive requests from multiple clients.
  2. A client can send requests to multiple services.
  3. A client can send requests to other clients.
  4. A service can respond to requests from multiple clients.

## How is a sequential service designed?

* 1. One object that responds to requests from clients
  2. Multiple objects that respond to requests from clients
  3. One subsystem that responds to requests from clients
  4. Multiple subsystems that respond to requests from clients

## How is a concurrent service designed?

* 1. One object that responds to requests from clients
  2. Multiple objects that respond to requests from clients
  3. One subsystem that responds to requests from clients
  4. Multiple subsystems that respond to requests from clients

## What is a database wrapper class?

* 1. A class that encapsulates a data structure (b) A class that encapsulates a database

1. A class that encapsulates the details of how to access data in a database
2. A class that encapsulates a relational table

## When designing an entity class as a relational table, which of the following is NOT true?

* 1. The relational table has multiple primary keys.
  2. The relational table has multiple foreign keys.
  3. The relational table has a primary key. (d) The relational table has a concatenated primary key.

## When mapping an aggregation hierarchy to a relational table, which of the following is NOT true?

* 1. The aggregate and part tables have different primary keys.
  2. The aggregate and part tables have the same primary key.
  3. The primary key of the aggregate table is a foreign key of the part table.
  4. The primary key of the part table is a foreign key of the aggregate table.

## When mapping a generalization/specialization relationship to a relational database, which of the following is NOT possible?

* 1. The superclass and each subclass are designed as relational tables.
  2. Only subclasses are designed as relational tables.
  3. The aggregate and part classes are designed as relational tables.
  4. Only the superclass is designed as a relational table.

# Chapter 16: Designing Service- Oriented Architectures

## What is a service-oriented architecture (SOA)?

* 1. A distributed software architecture consisting of multiple related services
  2. A distributed software architecture consisting of multiple autonomous services
  3. A distributed client/service architecture (d) A distributed software architecture

## Which of the following properties DOES NOT apply to a service?

* 1. Reusable (b) Discoverable

(c) Fixed (d) Autonomous

## In a SOA, which of the following is NOT true?

* 1. A client communicates with a specific service provided on a fixed server configuration.
  2. A client discovers and links to a service.
  3. Multiple clients communicate with a service.
  4. Standard protocols are provided to allow clients to communicate with services.

## What is an object broker?

* 1. An object that breaks into a system
  2. An object that sends requests to other objects
  3. An object that handles requests sent by other objects
  4. An object that mediates interactions between clients and services

## Why does a service register with a broker?

* 1. So that service requesters can discover it (b) So that a service can interrogate the broker

(c) So that the registry is up to date (d) So that the service can relocate

## When is it particularly useful to use the Broker Handle pattern in place of the Broker Forwarding pattern?

* 1. If the client only communicates with the service once
  2. If the client needs to have a dialog with the service
  3. If the client knows the type of service required but not the specific service
  4. If the client needs to provide the broker with a handle

## Yellow pages brokering is useful when a service requester:

* 1. Needs to discover the location of the service
  2. Knows the type of service required but not the specific service
  3. Knows the specific service required but not the type of service
  4. Needs to discover the broker

## What is a transaction?

* 1. Consists of two or more operations (b) Consists of one operation

1. Consists of two or more operations that are indivisible
2. Consists of two or more operations that are divisible

## What is a compound transaction?

* 1. The compound transaction is indivisible. (b) The compound transaction is atomic.

1. The compound transaction is decomposed into atomic transactions.
2. The compound transaction is decomposed into subatomic transactions.

## With a Negotiation pattern, which of the following is NOT true?

* 1. The client agent can propose a service.
  2. The service agent can offer a service in response to a client agent proposal.
  3. The client agent can request a service.
  4. The service agent can offer a service in response to a client agent request.

# Chapter 17: Designing Component-Based Software Architectures

### In a distributed component-based software architecture, which of the following statements is the most complete description of component deployment?

* 1. Component instances can be deployed to different nodes in a geographically distributed environment.
  2. Component instances can be deployed to different nodes in a geographically distributed environment before design.
  3. Component instances can be deployed to different nodes in a geographically distributed environment before implementation.
  4. Component instances can be deployed to different nodes in a geographically distributed environment after design and implementation.

### What does a component interface consist of?

* 1. The externally visible operations of a component (b) The operations provided by a component

(c) The operations required by a component (d) The operations that a component supports

### What does a component’s provided interface consist of?

* 1. The operations that a component must fulfill (b) The operations inside a component

(c) The operations that a component uses (d) The operations of a component

### What does a component’s required interface consist of?

* 1. The operations that a component must fulfill (b) The operations inside a component

(c) The operations that a component uses (d) The visible operations of a component

### What does a connector join?

* 1. The provided port of one component to the required port of another component
  2. The provided port of one component to the provided port of another component
  3. The required port of one component to the provided port of another component
  4. The required port of one component to the required port of another component

### What does a delegation connector join?

* 1. An outer provided port to an inner provided port
  2. An outer provided port to an inner required port
  3. An outer required port to an inner provided port
  4. An outer provided port to an outer required port

### What is broadcast message communication?

* 1. A message sent to several recipients (b) A message sent to a specific recipient

(c) A message sent to all recipients (d) A message sent to recipients who are members of a group

### What are the communication characteristics of subscription/notification?

* 1. A message sent to several recipients (b) A message sent to a specific recipient

(c) A message sent to all recipients (d) A message sent to recipients who have joined a group

### During application deployment:

(a) The application is executed. (b) Component instances are executed.

(c) Component instances are assigned to hardware nodes. (d) Component instances are instantiated.

### What is an advantage of localized autonomy in component-based design?

* 1. If a component goes down, other components can continue to execute.
  2. Components execute concurrently.
  3. Components are distributed. (d) Components communicate using messages.

# Chapter 18: Designing

**Concurrent and Real-Time Software Architectures**

### What is the difference between an active object and a passive object?

* 1. An active object controls a passive object.
  2. An active object does not have a thread of control; a passive object has a thread of control.
  3. An active object executes in a distributed system; a passive object executes in a centralized system.
  4. An active object has a thread of control; a passive object does not have a thread of control.

### What is an event-driven input task?

* 1. A task that executes every few seconds
  2. A task that controls other tasks
  3. A task that receives inputs from an external device when it generates interrupts
  4. A task that checks whether there is new input from an external device

### What is a periodic task?

* 1. A task that responds to each message it receives (b) A task that is activated by a timer event

1. A task that is activated by an external event
2. A task that is activated by an input event

### What is a demand-driven task?

* 1. A task that responds to each message it receives
  2. A task that is activated by an internal message or event from another task
  3. A task that is activated by an external event
  4. A task that is activated by an input event

### What is a control task?

* 1. A task that control other tasks (b) A task that executes a statechart

(c) A task that executes on demand (d) A task that controls I/O devices

### What is a user interaction task?

* 1. A task that interacts with I/O devices (b) A task that interacts with users

(c) A task that interacts with a user sequentially (d) A task that interacts with a user concurrently

### Which of the following is true for a Centralized Control architectural pattern?

* 1. Control is divided among various control components.
  2. It provides the overall control and sequencing of the system.
  3. It provides overall control by coordinating several control components.
  4. It provides overall control over various I/O objects.

### Which of the following is true for a Distributed Control architectural pattern?

* 1. Control is divided among various control components.
  2. It responds to multiple requests from client subsystems.
  3. It provides overall control by coordinating several control components.
  4. It provides distributed control over various I/O objects.

### Which of the following is true for a Hierarchical Control architectural pattern?

* 1. Control is divided among various control components.
  2. It provides overall control over several client subsystems.
  3. It provides overall control by coordinating several control components.
  4. It provides overall control over various I/O objects.

1. Which of the following is NOT a case of event synchronization?

(a) External event (b) Internal event (c) Timer event (d) User event

# Chapter 19: Designing Software Product Line Architectures

### What is a software product line (SPL)?

* 1. A family of systems with some common components and some variable components
  2. An assembly line (c) A family of identical systems

(d) The software products marketed by a company

### What is an optional use case?

* 1. A use case with some optional steps (b) A use case that does not need to be developed

1. A use case that is required by some product line members but not others
2. A use case that can be chosen in place of a different use case in a SPL member

### What is a use case variation point?

* 1. A variable use case (b) A location in the use case at which change can occur

(c) An alternative use case (d) A location in the use case where an alternative path can start

### What is a SPL feature?

* 1. A requirement or characteristic that is provided by one or more SPL members
  2. A marketing need (c) A class provided by the SPL

(d) A SPL use case

### What is a SPL feature group?

* 1. A collection of features
  2. A group of features with a particular constraint on their usage in a SPL member
  3. A group of mutually exclusive features
  4. A group of optional features with a particular constraint on their usage in a SPL member

### What is a kernel class in a SPL?

(a) An entity class in the SPL (b) A SPL class that stores essential data

(c) A class that is required by all members of the SPL (d) An external class to the SPL

### What two categories of stereotypes are used in modeling SPL classes?

* 1. Kernel and optional stereotypes
  2. Optional and variant stereotypes
  3. Common and variant stereotypes
  4. Reuse and application role stereotypes

### How are feature conditions used in a SPL state machine?

* 1. A guard condition (b) A condition that is True or False

1. To identify if a feature is selected or not in the state machine
2. To allow state machine inheritance

### What is a kernel system in a SPL?

* 1. A member of the SPL only composed of kernel classes
  2. A member of the SPL composed of kernel classes and possibly some default classes
  3. A member of the SPL composed of kernel classes and possibly some optional classes
  4. A member of the SPL composed of kernel classes and possibly some entity classes

### What does the SPL software architecture describe?

* 1. The software inside a family of buildings (b) The structure of a client/server product family

1. The overall structure of the software product line
2. The software product line classes and their relationships

# Chapter 20: Software Quality Attributes

### What do software quality attributes address?

* 1. Software functional requirements
  2. Software nonfunctional requirements
  3. Software performance requirements
  4. Software availability requirements

### What is maintainability?

* 1. The extent to which software is capable of being changed before deployment
  2. The extent to which software is capable of being changed after deployment
  3. The extent to which software is capable of being changed during development
  4. The extent to which software is capable of being changed after development

### What is modifiability?

* 1. The extent to which software is capable of being modified after deployment
  2. The extent to which software is capable of being modified after initial development
  3. The extent to which software is capable of being modified during and after initial development
  4. The extent to which software is capable of being changed before deployment

### What is testability?

* 1. The extent to which software is capable of being developed
  2. The extent to which software is capable of being tested before deployment
  3. The extent to which software is capable of being tested after deployment
  4. The extent to which the software is understood

### Traceability is the extent to which a product:

* 1. Can be traced back to products of previous phases
  2. Traced back to the requirements
  3. Traced forward to implementation
  4. Deployed to a hardware configuration

### What is scalability?

* 1. The extent to which an application can grow
  2. The extent to which the system is capable of growing after its initial deployment
  3. The extent to which the system is capable of growing during development
  4. The extent to which the system is capable of being scaled

### What is reusability?

* 1. The extent to which software implementation is reusable
  2. The extent to which software is capable of being reused
  3. The extent to which SPL technology can be introduced
  4. The extent to which the software is common among a program family

### Which of the following is not performance-related?

* 1. System response time (b) System throughput

(c) System availability (d) System capacity

### Which of the following is not addressed by a secure system?

* 1. System penetration (b) Denial of service

(c) System scalability (d) System authorization

### Which of the following system problems does availability address?

* 1. Denial of service (b) Single point of failure

(c) System throughput (d) System penetration

## Chapter 21: Client/Server Software Architecture Case Study

Hết

### Which of these can be found in Common Object Request Broker Architecture (CORBA) Choose the three correct answer

1. Interface definition language
2. Facilities
3. Object request broker
4. client and server stubs

### What does the component of ‘component diagram’ refer to?

1. An external part of the software system, like a library that must be be imported.
2. The basic parts of the software, which it could not run without.
3. an independent, encapsulated unit in the system.
4. a general name for a ‘part’ of the software system. It could be a method, variable, class, object, or grouping of any of these

### What are two major styles of SOAP request?

1. document style
2. envelope
3. RPC style
4. request-response

### Which of the following is a common messaging pattern in client/server relationships?

1. handshake
2. one-way.
3. solicit-response.
4. request-response.

### Divide-and-conquer is a suitable methodology for architecture design

1. True
2. False

Architecture design is about choosing the right single architecture style for a project

1. True
2. False

### The "+1" in Kruchten's 4+1 View Model refers to a scenario. What is a scenario?

* 1. a representation of a normal use case.
  2. a representation of the system-wide state.
  3. an unexpected use case of the software.
  4. one of the tools that is used to implement the software.

### Which of these statements about Component Diagrams is true?

* + 1. They clarify dependency relationships
    2. They do not show third-party libraries
    3. They give a dynamic view of the system
    4. They are useful for clarifying the artifacts that will be produced from development

### The purpose of the software design phase is to produce asoftware requirement specification.

1. True
2. False

A whole/part relationship is a generalization/specification hierachy

1. True
2. False

### Which of the following structrues describe the dynamic properties of software architecture?

1. Software management structure
2. Software deployment structure
3. Software code structure
4. Software runtime structure

Which of these UML diagrams are likely to be part of the process view? (select two)

1. Activity diagram
2. Class diagram
3. Sequence Diagram
4. State Diagram

### What of these is an advantages of event-based architectured?

1. Event generators and event consumers are loosely coupled
2. Events are processed with more efficiency
3. All interactions happen synchronously
4. Events trigger responses in a predictable manner