Answers to Key Terms

Suggested answers are provided below. These answers are presented top-down, left to right.

5. Actor 16. Concrete class
39. Use case 13. Class-scope attribute
40. Use-case diagram 2. Abstract operation
23. Object 20. Method
34. State 29. Polymorphism
11. Behavior 27. Overriding
24. Object class (class) 21. Multiple classification
12. Class diagram 6. Aggregation
25. Object diagram 15. Composition
26. Operation 36. State transition
18. Encapsulation 19. Event
17. Constructor operation 35. State diagram
30. Query operation 32. Sequence diagram
38. Update operation 3. Activation
7. Association 37. Synchronous message
31. Scope operation 33. Simple message
9. Association role 10. Asynchronous message
22. Multiplicity 4. Activity diagram
8. Association class 28. Package
1. Abstract class 14. Component diagram

Answers to Review Questions

1. A use case represents a sequence of related actions that have been initiated by an actor. An actor is an external entity that interacts with the system.

An extends relationship extends or adds to a use case by adding new relationships. For example, the Class Registration use case can function fine on its own; however, in certain circumstances, additional actions will need to be performed, such as registering a student for a special class, thus requiring an extension use case. However, when one use case requires the use of another use case while executing then a uses (or includes) relationship is appropriate.

An object, or object instance, refers to an individual object; an object class refers to a set of objects that share a common structure and a common behavior. Jacob James and Colby Michael are both object instances. Employee is an example of an object class.

An attribute is a property of an object; an operation is a function or a service that is provided by all the instances of a class.
An object’s state encompasses an object’s properties and the values those properties have. An object’s behavior represents how an object acts and reacts. The object’s behavior is dependent on the object’s state and the operation being performed.

An operation is a function or a service that is provided by all the instances of a class. The method is the actual implementation of the operation.

A query operation accesses the state of an object but does not alter the state; however, an update operation alters the state of an object.

An abstract class is a class that has no direct instances, but whose descendants may have direct instances. A concrete class is a class that can have direct instances.

A class diagram shows the static structure of an object-oriented model: the object classes, their internal structure, and the relationships in which they participate. An object diagram is a graph of instances that are compatible with a given class diagram.

An association is a relationship among instances of object classes. An aggregation is a stronger form of association, representing a part-of relationship between a component object and an aggregate object.

Generalization abstracts common features among multiple classes, as well as the relationships in which they participate. Aggregation is a part-of relationship between a component object and an aggregate object. While aggregation involves a set of distinct object instances, generalization relates object classes.

Aggregation refers to a part-of relationship between a component object and an aggregate object. Composition is a stronger form of aggregation, referring to a part object that belongs to only one whole object, and that lives and dies with the whole object.

Overriding by extension occurs when an operation inherited by a subclass is extended by adding some behavior; however, in contrast, overriding by restriction occurs when the inherited operation is restricted or tightened by a new operation.

As it relates to state diagrams, a state is a condition during the life of an object during which it satisfies some condition, performs some action, or waits for some event. An event is something that takes place at a certain point in time. An event changes an object’s state.

An event is something that takes place at a certain point in time; as a result of that event, an action occurs.

An entry action is performed upon entering a state; an exit action is performed when leaving a state.

A state diagram is a model of the states of an object and the events that cause the object to change from one state to another. A sequence diagram depicts the interactions among objects during a certain period of time.

A generic sequence diagram shows all possible sequences corresponding to all the scenarios of a use case. An instance sequence diagram shows the sequence for one scenario.
A synchronous message is a type of message in which the caller has to wait for the receiving object to finish executing the called operation before it can resume execution itself. An asynchronous message is a type of message in which the sender does not have to wait for the recipient to handle the message.

A sequence diagram depicts the interactions among objects during a certain period of time. An activity diagram shows the conditional logic for the sequence of system activities needed to accomplish a business process.

2. The object-oriented development life cycle involves developing an object through analysis, design, and implementation phases. As the model develops, its focus shifts from an external, abstract focus to one that is more detailed and internally focused. Object-oriented analysis focuses on what the new system should do, not how it will function. During this phase, a model of the real-world application is developed showing its important properties. During object-oriented design, the analyst will specify how the application-oriented model will be realized in the implementation environment. This necessitates research into the impact that the implementation environment will have on the design. The design phase consists of two subphases: system design and object design. During system design, the system designer proposes a system architecture. During object design, a design model is built by adding implementation features. During the implementation phase, the design is implemented via a programming language or DBMS.

3. The biggest difference between the models is that the object-oriented models build on each other, are consistent, and use a consistent notation. Models developed during structured analysis and design are weakly connected and lack a common representation.

4. An extends relationship is used when a use case requires the addition of new behaviors or actions. The use case is always performed, regardless of whether the extension use case is performed.

5. The include relationship is necessitated when a use case requires (uses) another use case while executing. In this situation, a specialized use case uses a generalized case.

6. Assume your university’s employment system hires new faculty and when eligible, promotes current faculty members to higher academic ranks. Hiring new employees and promoting current faculty are specialized use cases. As part of the hiring or promotion process, the individual’s vita is evaluated. So, the evaluate vita use case is an abstract case.

7. When an association has attributes, operations, or participates in relationships of its own, the association should be modeled as an association class.
8. Suggested answers are provided below.

**Relationship Examples**
**Class Diagrams**

![Binary relationship example diagram]

**Unary**

![Ternary relationship example diagram]

**Ternary**

9. Suggested answers are provided below.

**Role Names Added to Examples**
**Class Diagrams**

![Binary relationship example with role names]

**Unary**

![Ternary relationship example with role names]

**Ternary**
10. Suggested answers are provided below.

**Operations Added to Examples**

**Class Diagrams**

Unary

Ternary

11. A suggested answer is provided below.

**Generalization Example**
12. Since students will provide a variety of examples, their answers to this question will vary. The example provided in the previous answer does contain an abstract class.

13. Polymorphism means that the same operation may apply to two or more classes in different ways. One example is if the method that implements the computePay operation adds certain benefits, such as retirement, for the salaried employee.

14. One example of aggregation is a vehicle. A vehicle will have front and side air bags, an air conditioner, an engine, tires, possibly a trunk, and other components.

15. A suggested answer is provided below.

Game
State Diagram

[Diagram showing game states: In-Play, TimeOut, Over, transitions with labels: game starts, game ends, timeout called, game resumes]
Answers to Problems and Exercises

1. A suggested answer is provided below.

University Registration
Use Case Diagram

- Student
- Registration Clerk
- Instructor
- Bursar's Office
- Class registration
- Prereq courses not completed
- Registration for special class
- Receive housing payment
- Student billing
- Receive tuition payment
2. A suggested answer is provided below.

**University Registration**  
**Use Case Diagram (Modified)**
3. A suggested answer is provided below.

**Auto Rental Company**

**Use Case Diagram**

[Diagram showing use case interactions involving Customer, Reserve Car, Buy Insurance, Pick Up Car, Record Mileage, Return Car, Rental Clerk, Auction Car, Generate Bill, and Generate Bill for Corporate Customers.]
4. A suggested answer is provided below.

Faculty Tenure Review
Use Case Diagram
5. A suggested answer is provided below.

Auto Rental Company
Class Diagram

6. Suggested answers are provided below.

Company
Class Diagram
7. A suggested answer is provided below.

**Student Activity Class Diagram**

- **Student**
  - studentName
  - Address
  - phone
  - age
  - printActivityReport()

- **Activity**
  - name

- **Participates-in**
  - 1..* relationship
8. A suggested answer is provided below.

![Class Diagram](image)

**Sports Team**
**Class Diagram**
9. Suggested answers are provided below.

**Stillwater Antiques Use Case Diagram**

- **Buy Item**
  - **Buyer**
  - **Client**

- **Sell Item**
  - **Seller**

**Stillwater Antiques Class Diagram**

- **Sale**
  - `commissionPaid`
  - `sellingPrice`
  - `salesTax`
  - `dateSold`
  - `sellItem(Client, Item)`

- **Item**
  - `itemNumber {key}`
  - `description`
  - `askingPrice`
  - `condition`
  - `comments`

- **Client**
  - `clientNumber {key}`
  - `clientName`
  - `clientAddress`

- **Purchase**
  - `purchaseCost`
  - `datePurchased`
  - `condition`
  - `buyItem(Client, Item)`

- `1 buyer`
- `1 seller`
10. Suggested answers are provided below.

H. I. Topi School of Business
Use Case Diagram
11. A suggested answer is provided below.

**Nonprofit Organization**

*Class Diagram*

```
Person
  ssn
  name
  address
  phone

Employee
  dateHired

Volunteer
  skill

Donor
  findTopTen()

Item
  name
  price
```
12. A suggested answer is provided below.

Bank Account
Class Diagram

Account
acctNo
dateOpened
balance

mailStatement(customer)
<<abstract>> computeBalance()

{disjoint, complete}

Checking
serviceCharge
computeBalance()

Savings
interestRate
computeBalance()

Loan
interestRate
payment
computeBalance()
13. Suggested answers are provided below.

Registration and Title
Class Diagram
Registration and Title
Nested State Diagram

On Sale

- Receive Car / Prep Car
  - Prep
    - entry: Update Inventory
    - exit: Wash Car

- Prep Done / Move to Lot
  - Test
    - Regaining any dents incurred during shipping, fixing and paint problems, and washing the car.

- For
  - Customer Makes Offer
  - Offer Rejected

- Sale
  - do: Negotiate
    - Offer Accepted
      - Finalize Sale

- Car is on sale in showroom or lot, but no offers have been made. Occasionally the car is washed or taken out for a test drive by a customer.

- When in this state, the car cannot be sold to any other customer until or if the offer is rejected.
14. A suggested answer is provided below.
15. This is a good exercise to assign in conjunction with Problems and Exercises 3 and 14. Your students will need to make several assumptions about this exercise. For their scenario, your students should decide whether the customer is a corporate or individual customer. If the customer is a corporate customer, then most likely the corporate customer will have information on file with the rental company, necessitating an object on the sequence diagram. You should ask your students to provide any assumptions that they make to you in written form. In its simplest interpretation, the diagram will have reservation, reservation entry, and available car objects appearing on the sequence diagram.

16. This is a good exercise for comparing and contrasting data flow diagrams with activity diagrams. Activity diagrams show the conditional logic for the sequence of system activities needed to accomplish a business process; data flow diagrams illustrate the flow of data between external entities and the processes and data stores within a system. In contrast to data flow diagrams, activity diagrams associate an activity with the organizational unit, show time (without a time scale), and sequence. A suggested answer is provided below.

**Hiring Process**

**Activity Diagram**

```
Activity Diagram

Human Resources

- Preview Applicant's Qualifications
- Choose Candidates
- Schedule Interview
- Hire/Reject Candidate
- Notify Interested Parties

Department

- Post Job Opening
- Interview Candidates
- Provide Evaluation

Make Hiring Decision
```