

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CS6502-OBJECT ORIENTED ANALYSIS AND DESIGN**

**UNIT-I
UML DIAGRAMS**

Introduction to OOAD – Unified Process - UML diagrams – Use Case – Class Diagrams– Interaction Diagrams – State Diagrams – Activity Diagrams – Package, component and Deployment Diagrams.

PART-A

1. What is Object-Oriented Analysis?

During object-oriented analysis there is an emphasis on finding and describing the objects or concepts in the problem domain. For example, in the case of the flight information system, some of the concepts include Plane, Flight, and Pilot.

2. What is Object-Oriented Design?

During object-oriented design (or simply, object design) there is an emphasis on defining software objects and how they collaborate to fulfill the requirements. The combination of these two concepts shortly known as object oriented analysis and design.

3. What is Object-Oriented Analysis and Design? APRIL/MAY-2011

During object-oriented analysis there is an emphasis on finding and describing the objects or concepts in the problem domain. For example, in the case of the flight information system, some of the concepts include Plane, Flight, and Pilot.

During object-oriented design (or simply, object design) there is an emphasis on defining software objects and how they collaborate to fulfill the requirements. The combination of these two concepts shortly known as object oriented analysis and design.

4. What is Analysis and Design?

Analysis emphasizes an investigation of the problem and requirements, rather than a solution. Design emphasizes a conceptual solution (in software and hardware) that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects.

5. Define Design Class Diagrams

A static view of the class definitions is usefully shown with a design class diagram. This illustrates the attributes and methods of the classes.

6. What is the UML? MAY/JUNE 2012

The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems.

7. What are the three ways and perspectives to Apply UML?

Ways - UML as sketch, UML as blueprint, UML as programming language Perspectives- Conceptual perspective, Specification (software) perspective, Implementation (Software) perspective.

10. Define Requirements and mention its types.

Requirements are capabilities and conditions to which the system and more broadly, the project must conform.

1. Functional
2. Reliability
3. Performance
4. Supportability

11. What are Actors?

An actor is something with behavior, such as a person (identified by role), computer system, or organization; for example, a cashier.

12. What is a scenario?

A scenario is a specific sequence of actions and interactions between actors and the system; it is also called a use case instance. It is one particular story of using a system, or one path through the use case; for example, the scenario of successfully purchasing items with cash, or the scenario of failing to purchase items because of a credit payment denial.

13. Define Use case.

A use case is a collection of related success and failure scenarios that describe an actor using a system to support a goal. Use cases are text documents, not diagrams, and use-case modeling is primarily an act of writing text, not drawing diagrams.

14. What are Three Kinds of Actors?

Primary actor, Supporting actor, offstage actor.

15. What Tests Can Help Find Useful Use Cases?

1. The Boss test
2. The EBP test
3. The Size test

16. What are Use Case Diagrams?

A use case diagram is an excellent picture of the system context; it makes a good context diagram that is, showing the boundary of a system, what lies outside of it, and how it gets used. It serves as a communication tool that summarizes the behavior of a system and its actors.

17. What are Activity Diagrams?

A diagram which is useful to visualize workflows and business processes. These can be a useful alternative or adjunct to writing the use case text, especially for business use cases that describe complex workflows involving many parties and concurrent actions.

19. What is an object?

An object is a combination of data and logic; the representation of some real-world entity.

20. What is the main advantage of object-oriented development?

- High level of abstraction

- Seamless transition among different phases of software development
- Encouragement of good programming techniques.
- Promotion of reusability.

21. What is Object Oriented System development methodology?

Object oriented system development methodology is a way to develop software by building self-contained modules or objects that can be easily replaced, modified and reused.

22. Distinguish between method and message in object.

Method Message

- i) Methods are similar to functions, procedures or subroutines in more traditional programming languages. Message essentially are non-specific function calls.
- ii) Method is the implementation. Message is the instruction.
- iii) In an object oriented system, a method is invoked by sending an object a message. An object understands a message when it can match the message to a method that has the same name as the message.

23. What is an activity diagram?

A UML activity diagram shows sequential and parallel activities in a process. They are useful for modeling business processes, workflows, data flows, and complex algorithms. Basic UML activity diagram notation illustrates an action, partition, fork, join, and object node. In essence, this diagram shows a sequence of actions, some of which may be parallel. Most of the notation is self-explanatory; two subtle points:

- once an action is finished, there is an automatic outgoing transition
- the diagram can show both control flow and data flow

24. What is meant by interaction diagram?

The term *interaction diagram* is a generalization of two more specialized UML diagram types; both can be used to express similar message interactions:

- . Collaboration diagrams
- . Sequence diagrams

25. What is meant by link?

A **link** is a connection path between two objects; it indicates some form of navigation And visibility between the objects is possible . More formally, a link is an instance of an association. For example, there is a link or path of navigation from a Register to a Sale, along which messages may flow, such as the make 2 Payment message.

26. What is meant by Messages?

Each message between objects is represented with a message expression and small arrow indicating the direction of the message. Many messages may flow along this link. A sequence number is added to show the sequential order of messages in the current thread of control.

27. How to create an instance?

Any message can be used to create an instance, but there is a convention in the UML to use a message named *create* for this purpose. If another (perhaps less obvious) message name is used, the message may be annotated with a special feature called a UML stereotype, like so:

«*create*».

The *create* message may include parameters, indicating the passing of initial values. This indicates, for example, a constructor call with parameters in Java.

28. List the relationships used in class diagram? APRIL/MAY-2011

Generalization(class to class)

Association (object to object)

Aggregation (object to

object)

Composition (object to object)

29. Define Events, States, and Transitions. APRIL/MAY-2011

An event is a significant or noteworthy occurrence.

For example:

A telephone receiver is taken off the hook.

A state is the condition of an object at a moment in time—the time between events.

For example:

- A telephone is in the state of being "idle" after the receiver is placed on the hook and until it is taken off the hook.

A transition is a relationship between two states that indicates that when an event occurs, the Object moves from the prior state to the subsequent state.

For example:

- When the event "off hook" occurs, transition the telephone from the "idle" to "active" state.

30. What is meant by State chart Diagrams?

A UML state chart diagram, illustrates the interesting events and states of an object, and the behavior of an object in reaction to an event. Transitions are shown as arrows, labeled with their event. States are shown in rounded rectangles. It is common to include an initial pseudo-state, which automatically transitions to another state when the instance is created.

31. State chart Diagrams in the UP?

There is not one model in the UP called the "state model." Rather, any element in any model (Design Model, Domain Model, and so forth) may have a state chart to better understand or communicate its dynamic behavior in response to events. For example, a state chart associated with the *State* design class of the Design Model is itself part of the Design Model.

32. Utility of Use Case State chart Diagrams.

- Hard-coded conditional tests for out-of-order events
- Use of the *State* pattern (discussed in a subsequent chapter)
- disabling widgets in active windows to disallow illegal events (a desirable approach)
- A state machine interpreter that runs a state table representing a use case State chart diagram.

33. List out the types of Events.

- External event
- Internal event
- Temporal event

34. Define External event.

External event—also known as a system event, is caused by something (for example, an actor) outside our system boundary. SSDs illustrate external events. Noteworthy external events precipitate the invocation of system operations to respond to them.

- When a cashier presses the "enter item" button on a POS terminal, an external event has occurred.

35. Define internal event.

Internal event—caused by something inside our system boundary. In terms of software, an internal event arises when a method is invoked via a message or signal that was sent from another internal object. Messages in interaction diagrams suggest internal events.

- When a *Sale* receives a *make Line item message*, an internal event has occurred.

36. Define temporal event.

Temporal event—caused by the occurrence of a specific date and time or passage of time. In terms of software, a temporal event is driven by a real time or simulated-time clock.

- Suppose that after an *end Sale* operation occurs, a *make Payment* operation must occur within five minutes, otherwise the current sale is automatically purged.

PART- B**1. Explain about Unified process phases. APRIL/MAY-2011**

- Iterative Development
- UP Practices and Concepts
- The UP Phases and Schedule
- The UP Disciplines (was Workflows)
- The Agile UP
- The Sequential "Waterfall"

2. Explain about Use-Case Model and its Writing Requirements in Context. APRIL/MAY-2011

- Background
- Use Cases and Adding Value
- Use Cases and Functional Requirements
- Use Case Types and Formats
- Fully Dressed Example: Process Sale
- Relating use cases- Include, Exclude, Generalize
- Example with diagram-ATM, Library Management System etc

3. List out the components of Object-Oriented Analysis and Design.

- Applying UML and Patterns in OOA/D
- Assigning Responsibilities
- What Is Analysis and Design?
- What Is Object-Oriented Analysis and Design?
- An Example

-The UML

4. What is activity diagram? Explain about its applications briefly? **APRIL/MAY-2011**

- UML Activity Diagram Notation
- Guidelines for activity modeling
- Example –Next Gen Activity Diagram

5. Explain about Interaction Diagram Notation? **APRIL/MAY-2011**

- Sequence and Collaboration Diagrams
- Collaboration Diagram
- Sequence Diagram
- Common Interaction Diagram Notation
- Basic Collaboration Diagram
- Notation
- Basic Sequence Diagram Notation

7. Illustrate with an example, the relationship between sequence diagram and use cases.

APRIL/MAY-2011

Sequence diagram- definition,example diagram

Use Cases- definition,example diagram

8. Explain UML State Machine Diagrams and Modeling.

- Definition
- How to apply
- Example
- Process

9. What is operation of UML Deployment and Component Diagram? Draw the diagram for a banking application. **APRIL/MAY-2011**

- Deployment Diagram
- Component Diagram