



CS6502-OBJECT ORIENTED ANALYSIS AND DESIGN

Question bank

Part A

Unit-I

Introduction to OOAD

1. **What is Object-Oriented Analysis? Nov/Dec 2016**
2. **What is Object-Oriented Design?**
3. **What is Object-Oriented Analysis and Design? APRIL/MAY-2011**
4. **What is Analysis and Design?**
5. **Define Design Class Diagrams April/May 2016**
6. **What is the UML? MAY/JUNE 2012**
7. **What are the three ways and perspectives to Apply UML?**
8. **What is Inception? APIRAL/MAY-2011**
9. **What Artifacts May Start in Inception?**
10. **Define Requirements and mention its types.**
11. **What are Actors?**
12. **What is a scenario?**
13. **Define Use case.-**
14. **What are Three Kinds of Actors?**
15. **What Tests Can Help Find Useful Use Cases?**
16. **What are UseCase Diagrams?**
17. **What are Activity Diagrams?**
18. **List the relationships used in use cases? MAY/JUNE 2012**
19. **19.List out the steps for finding use cases.NOV/DEC 2012**
- 20 **What is an object?**



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

22. What is Object Oriented System development methodology?

23. Distinguish between method and message in object.

Part B

1. Explain about POS generation systems.

- The Next Gen POS System
- Architectural Layers and Case Study Emphasis
- Iterative Development and Iterative Learning

2. Define Inception. Explain about artifacts of Inception

- Inception: An Analogy
- What Artifacts May Start in Inception
- You Didn't Understand Inception When...

3. 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"

4. 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

5. 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



Elaboration

PART- A

- 1. What is Elaboration?**
- 2. What are the tasks performed in elaboration?**
- 3. What are the key ideas and best practices that will manifest in elaboration?**
- 4. What Artifacts May Start in Elaboration?**
- 5. What are the key ideas for Planning the Next Iteration?**
- 6. What is a Domain Model? APIRAL/MAY-2011**
- 7. How the domain model is illustrated?**
- 8. Why Call a Domain Model a "Visual Dictionary"?**
- 9. What are the elements not suitable in a domain model?**
- 10. 10 What are Conceptual Classes?**
- 11. How to Create a Domain Model?**
- 12. How to Find Conceptual Classes?**
- 13. Mention some Conceptual Class Category.**
- 14. Why Should We Avoid Adding Many Associations?**
- 15. How to Name an Association in UML?**
- 16. What is Aggregation? APRIL/MAY-2011**
- 17. What is composition? APRIL/MAY-2011**
- 18. Mention the guidelines that suggest when to show aggregation.**
- 19. What is an activity diagram?**

Part B

- 1. Write briefly about elaboration and discuss the differences between Elaboration and Inception with examples.**



- Inception and Elaboration
- Planning the Next Iteration

2. Illustrate the concept of Domain model with examples. **APRIL/MAY-2011**

- Definitions
- Guidelines for creating domain model -Examples

3. Explain the guide lines for finding Conceptual Classes with neat diagrams

- Three Strategies
- Find and Draw Conceptual Classes

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 Aggregations and compositions

- Definitions
- Identify Composition &Aggregations -
- Example: the Next Gen Domain Model

UNIT-III

System Sequence Diagrams

PART- A

1. What is meant by System Sequence Diagrams? **APRIL/MAY-2011**

2. What is meant by System Behavior?

3. What is meant by Inter-System SSDs?

4. Define System Events and the System Boundary.

5. How to Naming System Events and Operations?

6. What is meant by interaction diagram?

7. What is meant by link?

8. What is meant by Messages?



9. How to create an instance?
10. What is meant by Low Coupling?
11. What is meant by High COHESION?
12. Define Controller.
13. What is meant by CRC card?
14. What is meant by Pure Fabrication?
15. List the relationships used in class diagram? APRIL/MAY-2011

Part B

1. How to Adding New SSDs and Contracts? -New System Sequence Diagrams
 - New System Operations
 - New System Operation Contracts
2. 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
3. Design the Model and Creating Design Class Diagrams. -When to Create DCDs
 - Example DCD
 - DCD and UP
 - Domain Model vs. Design Model Classes -DCDs, Drawing, and CASE Tools
 - DCDs within the UP
4. What are concepts involved in domain refinement?
 - Generalization
 - Defining Conceptual Super classes and Subclasses
 - Class Hierarchies and Inheritance
 - Aggregation and Composition
 - Examples



5. Illustrate with an example, the relationship between sequence diagram and use cases.
APIRAL/MAY-2011

Unit-IV

GRASP

PART- A

- 1. How to Choosing the Initial Domain Object?**
- 2. How to Connecting the UI Layer to the Domain Layer?**
- 3. Mention the Interface and Domain Layer Responsibilities.**
- 4. Define patterns.**
- 5. How to Apply the GRASP Patterns?**
- 6. Define Responsibilities and Methods.**
- 7. Who is creator?**
- 8. List out some scenarios that illustrate varying degrees of functional cohesion.**
- 9. Define Modular Design.**
- 10. What are the advantages of Factory objects?**
- 11. Designing for Non-Functional or Quality Requirements.**
- 12. Abstract for Factory (GoF) for Families of Related Objects.**
- 13. What is meant by Abstract Class Abstract Factory?**
- 14. What is meant by Fine-Grained Classes?**
- 15. Define coupling. APIRAL/MAY-2011**

PartB

- 1. Explain Grasp: designing objects with responsibilities.**



- Responsibilities and Methods -
- Responsibilities and Interaction Diagrams
- Patterns

2. Explain GRASP: Patterns of General Principles in Assigning Responsibilities.

APIRAL/MAY-2011

- The UML Class Diagram Notation -
- Information Expert (or Expert) -
- Creator

- low coupling
- high cohesion
- controller
- object design and CRC CARDS

3. How to Determining the Visibility of the Design Model?

- Visibility between
- Objects -Visibility

4. Explain about Patterns for Assigning Responsibilities. -Polymorphism

- Pure Fabrication
- Indirection

- Protected Variations

5. Designing the Use-Case Realizations with GoF Design Patterns. **APRIL/MAY-2011**

- Analysis" Discoveries during Design: Domain Model -
- Factory

- Singleton

- Conclusion of the External Services with Varying Interfaces Problem 3 -
- Strategy

- Composite -Facade

Unit-V

UML state diagrams and modeling

PART- A

- 1. Define post condition.**
- 2. Define Attributes.**
- 3. When Are Contracts Useful?**



4. **Mention the Guidelines for Contracts.**
5. **What are Steps for Mapping Designs to Code?**
6. **Creating Class Definitions from DCDs.**
7. **What are the Benefits of Iterative Development?**
8. **Define Events, States, and Transitions. APRIL/MAY-2011**
9. **What is meant by State chart Diagrams?**
10. **State chart Diagrams in the UP?**
11. **Utility of Use Case State chart Diagrams.**
12. **List out the types of Events.**
13. **Define External event.**
14. **Define internal event.**
15. **Define temporal event.**

Part B

1. Explain the operation of Mapping Designs to Code. **APRIL/MAY-2011**
 - Programming and the Development Process
 - Mapping Designs to Code
 - Creating Class Definitions from DCDs -
 - Creating Methods from Interaction
 - Diagrams - Container/Collection Classes in
 - Code - Exceptions and Error Handling
 - Defining the Sale--makeLineItem
 - Method - Order of Implementation
 - Test-First Programming
2. Explain about Implementations.