

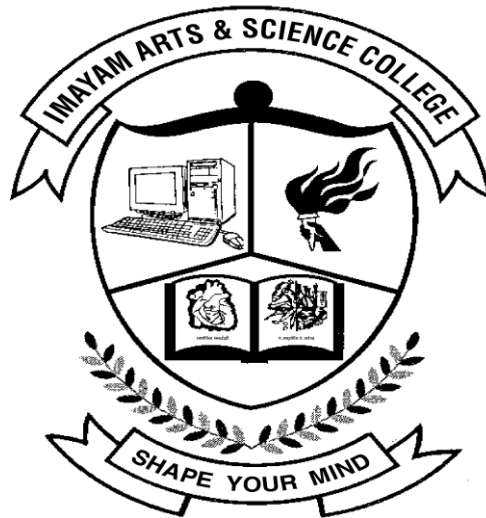
IMAYAM ARTS AND SCIENCE COLLEGE

KANNANUR, THURAIYUR

Department Of Computer Science & Applications

OOAD & UML

(P16CS21)



QUESTION BANK

SUBJECT HANDLED BY:

K.AKILA – I M.SC (CS)

CHOOSE THE CORRECT ANSWER

UNIT – I

1. Algorithm + Data Structure =
a) Functions b) Procedure c) Programs d) None
Ans: c

2. Each object has
a) attributes and methods b) data and functions c) both d) none
Ans: a

3. The unified modeling language originally was called
a) Unified Method b) Unified model c) Uniform model d) Uniform method
Ans: a

4. What is the programming style of the object oriented conceptual model?
a) Invariant relationships b) Algorithms c) Classes and objects d) Goals
Ans: c

5. Abstraction is classified into
a) 4 b) 3 c) 2 d) 1
Ans: a

6. Single inheritance, multiple inheritance and aggregation comes under
a) Modularity b) Typing c) Hierarchy d) None
Ans:c

7. The goal of OOD is to design the classes identified during
a) Analysis phase &user interface b) Design Phase c) Coding Phase. d) None

8. OMT consists of
a) 3 phases b) 2 phases. c) 4phases d) 5phases
Ans: c

9. OMT separates modeling into
a) 4parts. b) 5parts c) 2parts. d) 3parts
Ans: d

10. Each macro development process has its own micro development process
a) Micro development process

- b) Macro development process
- c) Both a&b
- d) None

Ans: a

UNIT – II

1. Which diagram in UML shows a complete or partial view of the structure of a modeled system at a specific time?
 - a) Sequence Diagram
 - b) Collaboration Diagram
 - c) Class Diagram
 - d) Object Diagram

Ans: d

2. Which of the following is a design pattern?
 - a) Behavioral
 - b) Structural
 - c) Abstract Factory
 - d) All of the mentioned

Ans: d

3. You want to avoid multiple inheritance. Which design pattern would you choose?
 - a) Abstraction-Occurrence Pattern
 - b) Player-Role Pattern
 - c) General Hierarchy Pattern
 - d) Singleton Pattern

Ans: b

4. Which pattern prevents one from creating more than one instance of a variable?
 - a) Factory Method
 - b) Singleton
 - c) Observer
 - d) None of the mentioned

Ans: b

5. UML is a graphical language with sets of
 - a) objects
 - b) semantics
 - c) rules and semantics
 - d) none

Ans: c

6. Qualifier is an
a) Association attribute b) Aggregation c) Semantics d) None
Ans: a
7. What does a simple name in UML class and objects consist of?
a) Letters b) Digits c) Punctuation characters and digits d) All of the above
Ans: d
8. OCL
a) Object Constraint Language b) Object Combine Language c) Both d) None
Ans: a
9. Which message is received so that the system moves to the Testing state, then the Transmitting state, before returning to the Running state?
a) signalStatus()
b) remoteControl()
c) reconfigure()
d) reportStatus()
Ans: d
10. If the system state is Shutdown then it can respond to which of the following message?
a) restart()
b) reconfigure()
c) powerSave()
d) all of the mentioned
Ans: d

UNIT - III

1. _____ is a reuse-based approach to defining, implementing, and composing loosely coupled independent components into systems.
a) Component-based software engineering
b) Component composition
c) Component model
d) Component interfaces
Ans: d

2. Service Oriented Architecture (SOA) is

- a) Strongly Coupled
- b) Loosely Coupled
- c) Strongly Cohesive
- d) Loosely Cohesive

Ans: b

3. How is SOA different from OO Architecture ?

- a) Strong coupling among objects
- b) Communications are prescriptive rather than being descriptive
- c) Data is separated from a service or behavior
- d) Data and methods are integrated into a single object

Ans: c

4. Which of the following is golden rule for interface design?

- a) Place the user in control
- b) Reduce the user's memory load
- c) Make the interface consistent
- d) All of the mentioned

Ans: d

5. Use cases _____

- a. Identify business processes
- b. Identify system goals
- c. Describe workflow
- d. Prioritize system procedures

Ans: b

6. Algorithmic and object-oriented are the two common ways for modeling _____

- a. Non-software Systems
- b. Software Systems
- c. Vocabulary of a System
- d. Client/Server System

Ans: b

7. UML is useful to _____ a system as it is or as we want it to be.

- a. Visualize
- b. Specify
- c. Document
- d. All of the above

8. _____ is an abstraction of a set of functions that the system performs.

- a. Class
- b. Interaction
- c. Use case
- d. Collaboration

Ans: c

9. The explanatory parts of the UML model are known as _____

- a. Behavioral things
- b. Grouping things
- c. Structural things
- d. Annotational things

Ans: d

10. _____ are used to create new building blocks from existing blocks.

- a. Tagged Values
- b. Stereotypes
- c. Constraints
- d. Diagrams

Ans: b

UNIT – IV

1. Set of inputs, execution preconditions and expected outcomes is known as a

- a) Test plan
- b) Test case
- c) Test document
- d) Test Suite

Ans: b

2. Which granularity level of testing checks the behavior of module cooperation?

- a) Unit Testing
- b) Integration Testing
- c) Acceptance Testing
- d) Regression Testing

Ans: b

3. Which of the following is not a user interface design process?
- a) User, task, and environment analysis and modeling
 - b) Interface design
 - c) Knowledgeable, frequent users
 - d) Interface validation

Ans: c

4. What incorporates data, architectural, interface, and procedural representations of the software?
- a) design model
 - b) user's model
 - c) mental image
 - d) system image

Ans: a

5. What combines the outward manifestation of the computer-based system , coupled with all supporting information that describe system syntax and semantics?
- a) mental image
 - b) interface design
 - c) system image
 - d) interface validation

Ans: c

6. How many axioms available in that?
- a) 1 b) 3 c) 2 d) 4

Ans: c

7. An axioms is a
- a) fundamental truth and rule b) derivation c) set of rules d) none

Ans: a

8. A system has a
- a) static part b) dynamic part c) static and dynamic part d) none

Ans: c

UNIT – V

1. Which web app attribute is defined by the statement:”A large number of users may access the WebApp at one time”?
 - a) Unpredictable load
 - b) Performance
 - c) Concurrency
 - d) Network intensiveness

Ans: c

2. Which web app attribute is defined by the statement:”The quality and aesthetic nature of content remains an important determinant of the quality of a WebApp”?
 - a) Availability
 - b) Data driven
 - c) Content sensitive
 - d) Continuous evolution

Ans: c

3. If the user queries a collection of large databases and extracts information from the webapp, the webapp is categorized under
 - a) Service oriented app
 - b) Database access app
 - c) Portal app
 - d) Data warehousing app

Ans: d

4. Which process model should be used in virtually all situations of web engineering?
 - a) Incremental Model
 - b) Waterfall Model
 - c) Spiral Model
 - d) None of the mentioned

Ans: a

5. Which analysis is a part of Analysis model of the web engineering process framework?
 - a) Content Analysis
 - b) Interaction Analysis
 - c) Functional Analysis
 - d) All of the mentioned

Ans: d

6. Which of the following statements are incorrect with reference to web-based systems?

Web-based systems

- a) should be unscalable
- b) must be able to cope with uncertain, random heavy demands on services
- c) must be secure
- d) are subject to assorted legal, social, and ethical scrutiny

Ans: a

7. What category of web-based system would you assign to electronic shopping?

- a) Informational
- b) Interactive
- c) Transaction-oriented
- d) Workflow-oriented

Ans: c

8. What category of web-based system would you assign to discussion groups?

- a) Collaborative work
- b) Online communities
- c) Web portals
- d) Workflow-oriented

Ans: b

9. W3C stands for

- a) World Wide Web Consortium
- b) World Wide Web Collaboration
- c) World Wide Web Community
- d) None of the mentioned

Ans: a

ANSWER THE FOLLOWING (2 MARKS)

UNIT – I

1. Define software development methodology:

A software development methodology is a series of processes that, if followed, can lead to development of an application.

2. Define UML:

The unified modeling language (UML) is a set of notations and conversations used to describe and model an application.

3. Define algorithm-centric and data- centric methodology:

In an algorithm-centric methodology, you think of an algorithm that can accomplish the task, then build data structures for that algorithm to use. In a data- centric methodology, you think how to structure the data, then build the algorithm around that structure.

4. Define object:

The term object means a combination of data and logic that represents some real world entity.

5. Define class hierarchy:

An object-oriented system organizes classes into a subclass-superclass hierarchy. Different properties and behaviours are used as the basis for making distinctions between classes and sub classes.

6. Define information hiding:

Information hiding is the principle of concealing the internal data and procedures of an object and providing an interface to each object in such a way as to reveal as little as possible about its inner workings.

7. Define inheritance:

Inheritance is the property of object-oriented systems that allows objects to be built from other objects. Inheritance allows explicitly taking advantage of the commonality of objects when constructing new classes.

8. Define polymorphism:

Poly means “many” and morph means “form”. Polymorphism means that the same operation may behave differently on different classes.

9. Define OOD:

The goal of object-oriented design (OOD) is to design the classes identified during the analysis phase and the user interface.

10. Define CBD:

Component-based development (CBD) is an industrialized approach to the software development process. Application development moves from custom development to assembly of prebuilt, pretested, reusable software components that operate with each other.

11. Define RAD:

Rapid application development (RAD) is a set of tools and techniques that can be used to build an application faster than typically possible with traditional methods.

UNIT – II

1. List out the things in UML:

There are four kinds of things in UML.

- Structural things
- Behavioural things
- Grouping things
- Realization things

2. Define interface:

An interface is a collection of operations. The declaration of interface look like a class with the keyword <interface>.

3. Define collaboration:

It defines an interaction. Collaboration have structures as well as behavioural dimensions. A collaboration is rendered as an ellipse with dashed line.

4. List out of the relationship in the UML:

There are four kinds of relationships in the UML.

- Dependency
- Association
- Generalisation
- Realization

5. Define use case:

It's a description of sequences of actions. A use case is used to structure the behavioural things in a model. A use case is rendered as an ellipse with solid lines.

6. What is meant by association:

An association is a structured relationship among classes that describes a set of links. An association is rendered as solid lines.

7. What is the use of booch methodology?

It's a widely used method.

It's used to analysis and design.

- Macro development process
- Micro development process

8. What is Jacobsan methodology?

Object oriented software engineering is developed by Ivar Jacobsan in 1992. It's the first object oriented methodology.

9. List out of the diagrams in booch methodology?

- Class diagram
- Object diagram
- State diagram
- Interaction diagram
- Module diagram
- Process diagram

UNIT – III

1. Define use case diagram?

- A use case describes how a user uses a system to accomplish a particular goal.
- A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize.

2. What is use case diagram in UML?

- A use case a list of actions or event steps typically defining the interactions between a role of an actor and a system to achieve a goal.
- A use case is a useful technique for identifying, clarifying and organizing system requirements.

3. List out of the use case diagram notations:

- Actors
- Use case
- Relationship
- System boundary

4. What is the advantages of OOA?

Focuses on data rather than the procedure as in structured analysis. It allows effective management of software complexity by the virtue of modularity.

5. Define transition:

A transition denotes a change in the state of an object. If an object is in a certain state when an event occurs the object may perform certain activities subject to specified conditions and change the state.

6. List out of the parts of states:

- Name
- Entry/Exit actions
- Internal transitions
- Sub-states

UNIT – IV

1. Define OOD:

After the analysis we can developed into an object oriented model using object oriented design.

The stages for object oriented design can be identify as,

- Definition of the system.
- Designing system architecture.
- Identification of the object in the system.
- Construction of design model.
- Specification of object interfaces.

2. Define design axioms:

- An axioms is a fundamental truth, rules. The axioms cannot be proven are derived.
- A therom is valid if it is referent axiom and detective steps are valid.

3. Types of the axioms:

There are two types of axioms.

- Axiom 1
- Axiom 2

4. Define system design:

It involves defining the system followed by designing the architecture of the system. A system has a static and dynamic part the static of the system is designed using a simple block diagram.

5. Define user interface layer:

The user interface layer consists of objects with which the user interfaces as well as the objects needed to manage or control the interface. The user interface layer is also called view layer.

6. Define business layer:

The business layer contains all the objects that represent the business (both data and behaviour). This is where the real objects such as order, customer, line item, inventory and invoice exist.

7. Define coupling:

Coupling is a measure of strength of association established by a connection from one object or software component to another. Coupling is a binary relationship.

8. List out of the coupling:

- Content coupling
- Common coupling
- Control coupling
- Stamp coupling
- Data coupling

9. Define cohesion:

Coupling deals with interactions between objects or software components. We also need to consider interactions within a single object or software component, called cohesion.

10. Define multiplicity:

Multiplicity specifies the range of allowable associated classes. It is given for roles within associations, parts within compositions, repetitions, and other purposes.

UNIT – V

1. Difference between structural and behavioral model.

Structural diagrams show the things that are in the system being modeled. They include the class diagram, component diagram, composite structure diagram, deployment diagram, object diagram, and package diagram. Behavior diagrams show how the system being modeled behaves.

2. What is behavioral model in UML?

UML Behavioral Diagrams depict the elements of a system that are dependent on time and that convey the dynamic concepts of the system and how they relate to each other. The elements in these diagrams resemble the verbs in a natural language and the relationships that connect them typically convey the passage of time.

3. What are the three types of modeling in UML?

The three types of modeling in UML are as follows:

- Structural modeling: - It captures the static features of a system. - It consists of the following diagrams: ...
- Behavioral modeling: - It describes the interaction within the system. ...
- Architectural modeling: - It represents the overall framework of the system.

4. What is behavior modeling training?

Behaviour modelling training (BMT) is a popular training intervention which focuses on changing behaviours on the job. BMT improves trainees' knowledge, skills, and desired actions on the job

5. What is UML modeling with examples?

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts.

6. What is behavioral modeling with example?

Behavior modeling can also be used by retailers to estimate customer purchases. For example, a retailer may examine the types of products that a customer purchases, both in-store and online, and the find out the likelihood that the customer will purchase a new product based on his previous records.

7. What is Structural Modeling?

Structural model represents the framework for the system and this framework is the place where all other components exist. Hence, the class diagram, component diagram and deployment diagrams are part of structural modeling. They all represent the elements and the mechanism to assemble them.

8. What is structural diagram in UML?

Structural diagram is an important part of the UML. It represents the static aspect of the system and the static parts of the diagrams are represented by classes, interfaces, objects, components and nodes. These diagrams show the things and different objects in a system being modeled.

9. What is architectural model?

Architectural model represents the overall framework of the system. It contains both structural and behavioral elements of the system. Architectural model can be defined as the blueprint of the entire system. Package diagram comes under architectural modeling.

FIVE MARK QUESTIONS

UNIT - I

1. Why an object orientation?
2. Explain inheritance
3. Explain polymorphism
4. Explain OOD
5. Explain CBD
6. Explain about object relationships and associations
7. What are the advantage of object-oriented development?

UNIT – II

1. Explain Booch methodology:
2. Explain about Rumbaugh methodology
3. Draw the following diagram for student
 - use case diagram
 - class diagram
4. Explain about Jacobson methodology
5. Explain about any two things in UML

UNIT – III

1. Explain use case diagram notations
2. Explain about benefits of use case diagram
3. Explain the advantages and disadvantages of OOA
4. Explain the advantages and disadvantages of structured analysis
5. Explain about use case diagram:

UNIT – IV

1. Explain OOD
2. Explain design axioms
3. Explain about corollary
4. Explain coupling
5. Explain about purpose of a view layer interface

UNIT – V

1. Explain quality assurance test.
2. Explain test plan.
3. Explain Myers's debuting principles.
4. Explain user satisfaction test.
5. Explain continuous testing.

TEN MARK QUESTIONS

UNIT - I

1. Explain about class hierarchy:
2. Explain about encapsulation and information hiding:
3. Write a brief notes on OOD:
4. Write a brief notes on object-oriented systems development:

UNIT – II

1. Write a brief notes on things in UML:
2. Draw the following diagram for Railway:
 - use case diagram
 - class diagram
 - state diagram
 - activity diagram
 - data flow diagram
3. Write a brief notes on Booch methodology and Rumbaugh methodology:

UNIT – III

1. Write a brief notes on use case diagram in UML:
2. Explain about States and state transitions:
3. Write a brief notes on structured analysis Vs object oriented analysis:

UNIT – IV

1. Briefly explain about design axioms.
2. Write a brief note on user interface layer.
3. Explain in detail about control business logic layer.

UNIT – V

1. Briefly explain about testing strategies.
2. Briefly explains about developing usability test plans and test cases for the vianet bank ATM system.