

P.G DEPARTMENT OF COMPUTER APPLICATIONS

**19PMC320 OOAD & UML
K1 LEVEL QUESTIONS**

Unit I

1. A _____ is a series of processes that leads to the development of an application.
a)software development methodology
b)Software System
c)Program
d)None
2. Algorithms + Data Structures
a)Architecture
b)Functions
c)Programs
d)None
3. Task of predicting correspondence
a)Correspondence
b)Validation
c)Correctness
d)Verification
4. Software development is _____ and always undergoing major change.
A.static
B. protected
C. dynamic
D. public
5. Software development methodology is a _____ of processes
A. group
B. series
C. set
D. unique set.
6. The traditional techniques view software as a collection of _____ and isolated data.
A. methods.
B. classes.

C. programs.

D. data.

7. The following formula is coined by_____ "algorithms+ data structures =programs"

A. Niklaus Wirth.

B. Booch

C. Ali Bahrami.

D. Balagurusamy.

8. Prototype that aids for incremental development of the ultimate software solution

a)Domain prototype

b)Analysis prototype

c)Horizontal

d) Vertical

9. A software system is a set of _____ for performing certain action on certain data.

A. programs.

B. procedures.

C. mechanisms.

D. methods.

10. In an object oriented environment, software is a collection of _____ objects.

A. discrete.

B. static.

C. dynamic.

D. primary.

11. In an object-oriented system everything is an _____ and each is responsible for itself.

A. entity.

B. object.

C. interface.

D. integer.

12. The _____ approach supports abstraction at the function level.

A. bottom-up

B. narrow

C. top-down

D. backtracking

13. _____ is an object-oriented language

A. C

B. Pascal

C. Java

D. FORTRAN

Unit II

14. Reusability is supported by _____ concept.

- A. method overloading.
- B. polymorphism
- C. inheritance.**
- D. inline function.

15. What is that concept in type theory in which a single name may denote objects of many different classes that are related by some common super class referred to _____

- a) Monomorphism
- b) Type Checking
- c) Polymorphism
- d) Generalization

16. The main advantage of an object-oriented system is that the class tree is _____ and can grow.

- A. static.
 - B. dynamic.**
 - C. parallel.
 - D. sequential.
- ANSWER: B

17. Most traditional development methodologies are either _____ centric or data centric.

- A. program.
- B. attribute
- C. algorithm**
- D. number

18. The term _____ was first formally utilized in the simula language.

- A. class.
- B. object.
- C. method.**
- D. data.

19. _____ are used to distinguish one type of object from another.

- A. methods.
- B. procedures.
- C. formulas.
- D. classes**

20. A single object is simply an _____ of a class.

- A. instance.**
- B. data.
- C. variables.
- D. methods.

21. Properties represents the _____ of an object.

- A. type.
- B. state.**
- C. kind.
- D. action.

22. In which of the following mechanisms, types of all variables and expressions are not known until runtime

- a) Strong Typing
- b) Weak Typing
- c) Static Binding/ early binding
- d) Dynamic Binding/ late binding**

23. Which of the following is/are advantage of using object oriented programming?

- a) Code Reusability
- b) Can create more than one instance of a class without interference
- c) Platform independent
- d) All of the mentioned**

Unit III

24. UML stands for _____.

- A. Unified Modeling Language.**
- B. Union Method Language.
- C. Unified Method Language.
- D. Union Member Language.

25. The heart of the UA is _____.

- A. Booch.
- B. Rumbough.
- C. Jacobson's Use case**
- D. Ali.

26. _____ is a structural relationship that describes a set of links, a link being a connection among objects

- a) **Association**
- b) Dependency
- c) Generalization
- d) None

27. Analysis process which involves analyzing and designing of system from an object-oriented programming is termed as

- A. Object Analysis
- B. Object Oriented Analysis**
- C. Overall Oriented Analysis
- D. System Analysis

28. Activity diagram, use case diagram, collaboration diagram and sequence diagram are considered as types of

- A. non-behavioral diagrams

- B. non structural diagrams
- C. structural diagrams
- D. behavioral diagrams**

29. In Unified Modeling Language, diagrams which captures system static structure and provide foundation for other models is called

- a) deployment diagrams
- b) class diagrams**
- c) component diagrams
- d) object diagrams

30. What is multiplicity for an association?

- a) The **multiplicity at the target class end of an association is the number of instances that can be associated with a single instance of source class**
- b) The multiplicity at the target class end of an association is the number of instances that can be associated with a number instance of source class
- c) All of the mentioned
- d) None of the mentioned

31. Which of the following UML diagrams has a static view?

- a) Collaboration
- b) Use case**
- c) State chart
- d) Activity

32. Which among these are the rules to be considered to form Class diagrams?

- a) Class symbols must have at least a name compartment**
- b) Compartment can be in random order
- c) Attributes and operations can be listed at any suitable place
- d) None of the mentioned

Unit IV

33. Kind of diagrams which are used to show interactions between series of messages are classified as

- a. activity diagrams
- b. state chart diagrams
- c. collaboration diagrams**
- d. object lifeline diagrams

34. Interaction Diagram is a combined term for

- a) Sequence Diagram + Collaboration Diagram**
- b) Activity Diagram + State Chart Diagram
- c) Deployment Diagram + Collaboration Diagram
- d) None of the above

35. Diagrams which are used to distribute files, libraries and tables across topology of hardware are called

- a) **deployment diagrams**
- b) use case diagrams
- c) sequence diagrams
- d) collaboration diagrams

36. What does a component diagram consists of

- a) Components, their Relationship to the environment
- b) Packages and dependency
- c) Internal structure
- d) a, b
- e) a, c

37. For showing how several objects collaborate in single use case, which one of the following OOAD artifacts is the MOST useful?

- a. **Interaction Diagrams**
- b. Activity Diagrams
- c. Package Diagrams
- d. State Diagrams

38. On a state diagram, a guard condition is:

- e) Represented as a rectangle with rounded corners
- f) Represented as a dashed line
- g) Represented as an ellipse
- h) **Represented in Square bracket**

Unit V

39. Model management view is shown by

- a) **Package diagram**
- b) Deployment diagram
- c) Implementation diagram
- d) All the above

40. Elements of extensibility constructs

- a) Constraints
- b) Stereo types
- c) Tagged values
- d) **All the above**

41. Constraints are shown as expression strings enclosed in braces

- a) **True**

b) False

42. _____ diagrams are used to illustrate data structures, and the static snapshots instances of the things found in the class diagrams.

a. use case

b. Object

c. Collaboration

d. Sequence

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

a. Tagged Values

b. Stereotypes

c. Constraints

d. Diagrams

44. The association stereotype «Extends» indicates _____

a. Delegation of part of a task to another use case

b. The target use case is a subprocess of the source use cases

c. A specialized form of a use case

d. A deviation from the UML standard

P.G DEPARTMENT OF COMPUTER APPLICATIONS

19PMC320 OOAD & UML K2 LEVEL QUESTIONS

Unit I

1. What is Object Oriented Analysis and Design?

It's a structured method for analyzing, designing a system by applying the object-orientated concepts, and develop a set of graphical system models during the development life cycle of the software.

2. What is Object Oriented Analysis?

Object-oriented analysis (OOA) is the process of analyzing a task (problem domain), to develop a conceptual model that can then be used to complete the task. It is the process of extracting the needs of the system and what the system must do to satisfy the user requirements.

3. Define object oriented methodology

Object-oriented methodology is a set of methods, models, and rules for developing systems.

4. 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.

5. What is a Software development process?

Software development process consists of Analysis, design, implementation, testing & refinement to transform users' need into software solution

6. Difference between verification and validation.

Verification means "am I building the product right" ? Begin after specification accepted Validation means "am I building the right product "? Subjective - is specification appropriate? Uncover true users' needs, therefore establish proper design? Begins as soon as project starts

7. What is an object?

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

8. Difference between object oriented approach and traditional approach.

Traditional approach views software as a collection of programs (or functions) and isolated data. A program can be defined as **Algorithms + Data Structures = Programs**. Object-oriented systems development is a way to develop software by building self-contained modules or objects that can be easily replaced, modified, and reused.

9. What is Unified approach?

The unified approach (UA) is a methodology for software development. The UA, based on methodologies by Booch, Rumbaugh, and Jacobson, tries to combine the best practices, processes, and guidelines along with the Object Management Group's unified modeling language.

10. What is prototype?

Prototype is a system or partially complete system that is built quickly to explore some aspects of the system requirements. It is not intended as the final system.

11. What is Component-based development (CBD)

Application development moves from custom development to assembly of pre-built, pre-tested, reusable software components that operate with each other. A CBD developer can assemble components to construct a complete software system.

12. What is Rapid Application Development (RAD)

RAD is a set of tools and techniques that can be used to build an application faster than typically possible with traditional methods. It does not replace SDLC but complements it, since it focuses more on process description and can be combined perfectly with the object oriented approach.

Its task is to build the application quickly and incrementally implement the user requirements design through tools such as visual basic, power builder, etc.

13. Write the benefits of iterative Development

Software development and all of its activities including testing are an iterative process. Therefore, it can be a costly affair if we wait to test a product only after its complete development. Here incremental testing comes into picture wherein the product is tested during various stages of its development

Unit II

14. Define OMT modeling.

OMT modelling encompasses Object model, dynamic model and functional model

15. Define Booch methodology

The Booch methodology prescribes a **Macro development process** and a **Micro development process**. The Macro development process serves a controlling framework for the micro process and can take weeks or even months. **Primary concern** is technical management of the system.

16. What is a micro development process

Each Macro development process has its own micro development processes. It is a description of day to day activities by a single or small group of s/w developers. It consists of following steps : Identify classes & objects, Identify Class & object semantics, Identify class & object relationships, Identify class & object interfaces & implementation

17. What is Object Oriented Business Engineering(OOBE)

OOBE is Object modeling at enterprise level. Use cases are the central vehicle for modeling, providing traceability throughout the software engineering processes.

18. What is Objectory?

OOSE also called objectory is a method of object oriented development with **specific aim** to fit the development of large, real-time systems

19. What Is A Pattern?

A pattern is an abstract solution to a commonly occurring problem in a given context. A pattern describes a problem which occurs over and over again in our environment, and then describes the core of a solution to that problem, in such a way that we can use this solution a million times over, without ever doing it the same way twice.

20. What Is An Antipattern?

Documents unsuccessful attempts at providing solutions to certain recurring problems but includes reworked solutions that are effective.

21. What Is Framework?

Frameworks are partially completed software systems that may be targeted at a specified type of application.

Unit III

22. Define Comment on models

A model is a representation in a certain medium of something in the same or another medium. A model is an abstract representation of a system, constructed to understand the system prior to building or modifying it.

23. What is a view?

A view is simply a subset of UML modeling constructs that represents one aspect of a system. At the top level, views can be divided into three areas: structural classification, dynamic behavior, and model management

24. What is UML?

The Unified Modeling Language (UML) is a language for specifying, constructing, visualizing, and documenting the software system and its components. It is a graphical language with sets of rules and semantics.

25. What Do You Meant By Static And Dynamic Modeling?

Static modeling is used to specify structure of the objects that exist in the problem domain. These are expressed using class, object and USECASE diagrams.

But Dynamic modeling refers representing the object interactions during runtime. It is represented by sequence, activity, collaboration and statechart diagrams.

26. What is Class Diagram?

Class diagrams are the main building blocks of every object oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration.

27. Define an Object diagram

An **Object Diagram** can be referred to as a screenshot of the instances in a system and the relationship that exists between them. Since object diagrams depict behaviour when objects have been instantiated, we are able to study the behavior of the system at a particular instant. Object diagrams are vital to portray and understand functional requirements of a system.

28. What are Instances?

Instance is a concrete manifestation of an abstraction to which a set of operations can be applied and which has a state that stores the effects of the operations. Instances and objects are almost same. Instances don't stand alone; they are tied to an abstraction.

Instances can be of classes, components, nodes, use cases, and associations

29. Define Qualifier

Qualifiers are the properties of binary associations and are an optional part of association ends. A qualifier holds a list of association attributes, each with a name and a type. A qualifier is visually represented as a rectangle attached to the qualified end of the association relationship. Qualifiers identify subsets of related instances in association navigations; they provide a model of indices or keys for association ends.

30. List the types of Relationships?

There are different kinds of relationships: Dependencies, Generalization, and Association.

31. Define Dependencies

Dependencies are relationships between two entities that a change in specification of one thing may affect another thing. Most commonly it is used to show that one class uses another class as an argument in the signature of the operation.

32. What is generalization?

Generalization is relationships specified in the class subclass scenario, it is shown when one entity inherits from other.

33. Define Associations

Associations are structural relationships that are: a room has walls, Person works for a company. Aggregation is a type of association where there is a has a relationship, That is a room has walls, Also if there are two classes room and walls then the relationship is called a association and further defined as an aggregation.

34. What is aggregation?

Aggregation is a special kind of association that specifies a whole/part relationship between the aggregate (whole) and a component part.

35. What is Use case modeling?

A use-case model is a model of how different types of users interact with the system to solve a problem. It describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in satisfying these goals. A use-case model consists of a number of model elements such as use cases, actors and the relationships between them. It is used to graphically represent a subset of the model to simplify communications.

Unit IV

36.What is a State diagram?

State Diagram represents different states that objects in the system undergo during their life cycle.

37.Define a Sequence diagram

Sequence diagram is used to explore logic of complex operations, function or procedure. In this diagram, sequence of the interactions between the objects is represented step by step.

38.What is a Collaboration diagram

Collaboration diagram groups together the interaction between different objects.

39. Define an Activity diagram

Activity diagram gives detail view of the business logic. It captures the work flow of a system

40. Differentiate an Activity diagram from a Sequence Diagram?

- Activity diagram: captures the process flow. They are used for functional modeling
- Sequence diagram: tracks the interaction between the objects. They are used for dynamic modelling

41. Define a component

Components are the high-level reusable pieces out of which systems can be constructed.

42. Define Component diagram.

A component is a physical unit of implementation with well-defined interfaces that is intended to be used as a replaceable part of a system. Each component embodies the implementation of certain classes from the system design. A component diagram is a graph of the design components connected by dependency relationships. They are used to illustrate the structure of the system

43. Comparison between sequence and Collaboration Diagram.

A Sequence diagram is an interaction diagram that details about the operation that is carried out. Sequence diagram captures the interaction between the objects in the context of collaboration. Sequence diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time.

Collaboration Diagram represents the interaction of the objects to perform the behavior of a particular use case or a part of use case.

The designers use the Sequence diagram and Collaboration Diagrams to define and clarify the roles of the objects that perform a particular flow of events of a use case.

44. Define Event

An event is a noteworthy occurrence that has a location in time and space. It occurs at a point in time; it does not have duration.

45. Define a state

A state represents a discrete, continuous segment of time wherein the object's behavior will be stable. The object will stay in a state until it is stimulated to change by an event

46. Define a transition

A transition is a relationship between two states indicating that an object in the first state will perform certain actions and enter the second state when a specified event occurs and specified conditions are satisfied

47. Differentiate between Component and Deployment Diagram.

Component diagram shows you how different elements of your system have been grouped together (into assemblies / dlls etc) - and the link between these components. A Deployment diagram takes you one step further and describes on which hardware element do the component reside.

48. Define Collaboration.

Collaboration is a collection of named objects and actors with links connecting them. They collaborate in performing some task.

Unit V**49. Define Model management**

Model management consists of packages (including special kinds of packages) and dependency relationships among packages.

50. Define a Package.

Packages contain top-level model elements, such as classes and their relationships, state machines, use case graphs, interactions, and collaborations—anything not contained in some other element.

51. What is a model

A model is a package that encompasses a complete description of a particular view of a system. It provides a closed description of a system from one viewpoint. Model is tree-structured. The root package contains in itself nested packages that constitute the full detail of the system from the given viewpoint.

52. Define a UML Environment

UML models are used within an environment. The purpose and interpretation of the model are affected by the rest of the environment.

53. What Is Uml Architecture?

Takes care structural and behavioral aspect of a software system. Includes software usage, functionality, performance, reuse, economic and technology constraints.

54.What Are Uml Messages?

Specification of a communication.

55.Define Modeling In Uml And It Advantages?

- Model is a simplification of reality.
- Blueprint of the actual system.
- Specify the structural and behavior of the system.
- Templates for designing the system.
- Helps document the system.

56.What Are The Different Views In Uml?

Use Case view - Presents the requirements of a system.

- Design View - Capturing the vocabulary.
- Process View - Modeling the systems processes and threads.
- Implementation view - Addressing the physical implementation of the system.
- Deployment view - Model the components required for deploying the system.

57.Define Sdlc In Uml?

- SDLC is Software Development Life Cycle.
- SDLC of a system included processes like Use case driven, Architecture centric, Iterative and Incremental. This Life cycle is divided into phases.
- Phase is a time span between two milestones.
- The milestones are Inception, Elaboration, Construction, and Transition.
- Process Workflows that evolve through these phase are Business Modeling, Requirement gathering, Analysis and Design. Supporting Workflows are configuration, change management, and project management.

P.G DEPARTMENT OF COMPUTER APPLICATIONS
19PMC320 OOAD & UML
K3 LEVEL QUESTIONS

Unit I

1. Examine the advantages of object oriented development?
2. Why is reusability important? How does object oriented software development promotes reusability?
3. Describe the Software Development process
4. Why object orientation is needed?
5. Explain about CBD.
6. How will you build high quality software?
7. Explain about the RAD Model.

Unit II

8. List and explain the phases of OMT?
9. Explain about the Booch methodology
10. What are the primary symbols used in Data Flow Diagrams?
11. Explain the template for design patterns.
12. Examine Use Case driven approach
13. List the difference between good pattern and anti pattern.

Unit III

14. Explain the history of UML.
15. Examine the uses of Models and the levels of model.
16. What does a model mean and what are the aspects to be considered for a model.
17. Explain the kinds of relationships with their notations and functions.
18. Explain the Use case view and the relationship between Use cases
19. Examine associations with examples?
20. Discuss on generalization hierarchy with examples?
21. Examine aggregation with examples.

Unit IV

22. Describe the State chart and its components
23. Describe the different types of Events with examples?
24. Explain the activity diagrams with appropriate examples.
25. Describe an interaction and the components of an interaction diagram
26. Compare and contrast the sequence diagram with a collaboration diagram
27. Examine the Deployment view with suitable examples
28. Explain the implementation view with an appropriate diagram..
29. Explain the transitions and different types of transitions with suitable examples.
30. Examine the states and their types

Unit V

31. Discuss on the model management view
32. Explain about the packages and their dependencies
33. Examine the extensibility constructs.
34. Discuss on the semantic responsibilities
35. Examine modeling with tools
36. Examine the programming language responsibilities
37. Examine the Notational responsibilities
38. Describe the UML Environment.

P.G DEPARTMENT OF COMPUTER APPLICATIONS

19PMC320 OOAD & UML K4K5 LEVEL QUESTIONS

Unit I

1. Describe the Software Development process.
2. How can we build a high quality Software? Explain.
3. Describe the Object oriented Systems Development life cycle – a use case driven approach.
4. Discuss on the advantages of Object oriented Approach
5. Elucidate on Prototypes and their types

Unit II

6. Describe in detail about Rambugh's Methodology
7. Discuss on the Jacobson methodology.
8. Discuss on the Booch methodology
9. Discuss on the Template for patterns and capturing patterns
10. Elaborate on Frameworks
11. Differentiate between Patterns and Frameworks.
12. Discuss on the Unified approach with a block diagram

Unit III

13. Mention the primary goals in the design of the UML and Explain the basic concepts of Unified Modeling Language.
14. Elucidate on the nine UML diagrams.
15. Discuss on the Classifiers
16. Elaborate on the Class diagram with appropriate examples
17. Describe Association and generalization with appropriate examples.
18. Discuss on the object diagram with suitable examples
19. Discuss on models and the advantages of Modeling?
20. Discuss on the use cases with respect to online Airline reservation system
21. Explain the Use case diagram with account to Railway Reservation System

Unit IV

22. Elucidate on the State chart diagram with appropriate examples
23. Describe the events and it's types with suitable examples
24. Discuss on the different types of actions and states in a State machine with suitable example
25. Explain the Interaction diagrams with an example
26. Explain the sequential diagram with examples
27. Elaborate on the collaboration diagrams with appropriate examples.
28. Discuss on the Implementation view with appropriate example

Unit V

29. Discuss on the UML Environment in detail
30. Explain the Model management view in detail.
31. Elaborate on the Extension Mechanisms

32. Discuss on the Deployment view with appropriate example
33. Describe the Physical view with appropriate examples.