Object Model, Classes and Objects, Classification

1) What is object model?
   The object model encompasses the principle of abstraction, modularity, encapsulation, hierarchy, typing, concurrency and persistence. Object model characterizes the components of the physical or abstract system to be modeled by a programmed system.

2) What is Object Oriented Programming?
   OOP is a method of implementation in which programs are organized as co-operative collections of objects each of which represents an instance of some class and whose classes are all members of a hierarchy of classes united via inheritance relationships.

3) What is Object Oriented Design?
   OOD is a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and Dynamic models of the system under design.

4) What is Object Oriented Analysis?
   OOA is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain.

5) Define Abstraction?
   An Abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer.

6) Define Encapsulation?
   Encapsulation is the process of compartmentalizing the elements of an abstraction that constitute its structure the contractual interface of an abstraction and its implementation.

7) Define Modularity?
   Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules.
8)What is Typing?
Typing is the enforcement of the class of an object, such that objects of different types may not be interchanged, or at the most, they may be interchanged only in very restricted ways.

9)What is Persistence?
Persistence is the property of an object through which its existence transcends time (i.e., the object’s location moves from the address space in which it was created.

10)Define Object?
An object has state, behavior, and identity; the structure and behavior of similar objects are defined in their common class; the terms instance and object are interchangeable.

11)What is Class?
A class is a set of objects that share a common structure and a common behavior. A single object is simply an instance of a class.

12)What is cardinality?
The multiplicity in association is denoted by cardinality. There are three common kinds of cardinality across an association:
- One-to-One
- One-to-Many
- Many-to-Many

13)What is classification?
Classification is the means by which we are knowledge. In Object-Oriented Design classification allows us to expose the commonality within key abstractions and mechanisms, and eventually leads us to smaller and simpler architectures. Classification helps us to identify generalization, specialization, and aggregation hierarchies among classes.

14)What are the relationships in object-oriented languages?
The different relationships are:
- Association
- Inheritance
- Aggregation
- Using
- Instantiation
- Metaclass

15)Define Metaclass?
A Metaclass is a class whose instances are themselves classes. The primary purpose of a metaclass is to provide class variables (which are shared by all instances of
UNIT-2

Use case and Conceptual model

1. What is Use case?

A Use case is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor. Graphically, a use case is rendered as an ellipse.

2. What is an Actor?

An actor represents a coherent set of roles that users of use cases play when interacting with these use cases. Typically, an actor represents a role that a human, a hardware device, or even another system plays with a system.

3. How will you organize Use cases?

Use cases can be organized by grouping them in packages in the same manner classes are organized. Use cases can also be organized by specifying generalization, include, and extend relationships among them. These relationship are applied in order to factor common behaviors and in order to factor variants.

4. Define Use case diagram?

A Use case diagram is a diagram that shows a set of use cases and actors and their relationships.

5. What does Use case diagram contain?

* Use case
* Actors
* Dependency, generalization and association relationships

6. Mention the major elements required to form a conceptual model of language?

* UML’s basic building blocks
* Rules that dictate how those building blocks may be put together
* Some common mechanisms that apply throughout the UML

7. Name the building blocks of UML?

* Things
* Relationships
* Diagrams
8. What are kinds of things in UML?

* Structural things
* Behavioral things
* Grouping Things
* Annotational things

9. What are the kinds of Relationships?

* Dependency
* Association
* Generalization
* Realization

10. What is a diagram?

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships).

11. What are the different UML diagrams?

* Class diagram
* Object diagram
* Sequence diagram
* Collaboration diagram
* State chart diagram
* Activity diagram
* Component diagram
* Deployment diagram

12. What are the UML diagrams?

1. Class diagram
2. Use-case diagram
3. Behavior diagram
   3.1. Interaction diagram
      3.1.1. Sequence diagram
      3.1.2. Collaboration diagram
   3.2. Statechart diagram
   3.3. Activity diagram
4. Implementation diagram
   4.1. Component diagram
   4.2. Deployment diagram

13. What is Aggregation and Composition?
Aggregation is a form of association. A hollow diamond is attached to the end of the path to indicate aggregation.

Composition, also known as the a-part-of, is a form of aggregation with strong ownership to represent the component of a complex object. Composition is also known as a part-whole relationship. UML notation for composition is a solid diamond at the end of a path.

14. What is UML Sequence diagram?
Sequence diagrams are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

15. What is UML Collaboration Diagram?
Collaboration diagram represents collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

16. What is UML Statechart diagram?
A Statechart diagram shows the sequence of states that an object goes through during its life in response to outside stimuli and messages.

17. What is Activity diagram?
An activity diagram is a variation or special case of a state machine, in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations.

UNIT III
1. What is meant by a component diagram?
Component diagram shows a set of components and their relationships. Graphically, it is a collection of vertices and arcs.

2. What are the contents of Component diagram?
- Components
- Interfaces
- Dependency, Generalisation, Association and realization relationships
- Packages and subsystems

3. What are the common uses of Component diagram?
   1) To model source code
   2) To model executable releases
   3) To model physical databases
4) To model adaptatable systems

4. What is meant by a deployment diagram?
Deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. It is a collection of vertices and arcs.

5. What are the contents of Deployment diagram?
- Dependency and association relationships.
- Nodes.
- Packages and subsystems.

6. What are the common uses of Deployment diagram?
1) To model embedded system.
2) To model Client/Server systems
3) To model distributed systems.

7. Give 2 steps to model an embedded system?
   I. To identify devices and nodes that is unique to your system.
   II. To provide visual for unusual devices using the UML's extensibility mechanisms to define system specific stereo types with appropriate icons.

8. What do you mean by sequence diagram?
Sequence diagram is an interaction diagram that emphasizes the time-ordering of messages. Graphically, a sequence diagram that shows object arranged along x-axis and messages ordered in increasing time along the y-axis.

9. What do you mean by Collaboration diagram?
A collaboration diagram emphasizes the organization of objects that participate in object interactions. A collaboration diagram is formed by placing the objects that participate in the interaction as the vertices in a graph. Next, render the links that connect these objects as the arcs of this graph.

10. Difference between collaboration and sequence diagram?
   a) In collaboration diagram, there is a path to indicate how an object is linked to another.
   b) There is a sequence no: to indicate time-order of the message. Prefix the message with the no: increasing monotonically for each new message in the flow of control.

11. What are the common uses of interaction diagram?
   1) To model flows of control by time-ordering.
   2) To model flows of control by organization.
12 What is an activity diagram?
Activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine. Activity results in some action which is made up of executable atomic computations that result in a change in state of the system.

13. What are the essential components of a pattern?
- Name
- Problem
- Context
- Forces
- Solution
- Examples
- Resulting context
- Rationale
- Related Patterns
- Known uses

14. What is Pattern thumbnail?
A good pattern often begins with an abstract that provides a short summary or overview. This gives readers a clear picture of the pattern and quickly informs them of its relevance to any problems they may wish to solve. Sometimes such a description is called a thumbnail sketch of the pattern, or a pattern thumbnail.

15. Define Antipatterns
A pattern represents a "best practice." whereas an antipattern represents "worst practice" or a "lesson learned." Antipatterns come in two varieties:
- Those describing a bad solution to a problem that resulted in a bad situation.
- Those describing how to get out of a bad situation and how to proceed from there to a good solution.

16. Define pattern mining
The process of looking for patterns to document is called pattern mining or sometimes reverse architecting.

17. What are guidelines for capturing patterns?
- Focus on practicability
- Aggressive disregard of originality
- Nonanonymous review
- Writers workshops instead of presentations
- Careful editing

18. Define Framework?
A framework is a way of presenting a generic solution to a problem that can be applied to all levels in a development. Design and software frameworks are the most popular.

19. What are the major differences between design patterns and frameworks?
- Design patterns are more abstract than frameworks
- Design patterns are smaller architectural elements than frameworks
- Design patterns are less specialized than frameworks

20. What is pattern?
A pattern is an instructive information that captures the essential structure and insight of a successful family of proven solutions to a recurring problem that arises within a certain context and system of forces.

21. What are the characteristics of good pattern?
- It solves a problem
- It is a proven concept
- The solution is not obvious
- It describes a relationship
- The pattern has a significant human component

22. What is proto-pattern?
A “pattern in waiting,” which is not yet known to recur, is called proto-pattern.

23. What is generative patterns?
Generative patterns are pattern that describes a recurring problem and they can tell how to generate something and can be observed in the resulting system architectures.

24. What is nongenerative pattern?
Non generative patterns are static and passive. They describe recurring phenomena without necessarily saying how to reproduce them.

UNIT IV

1. What are the kinds of errors that you might encounter when you run a program?
- Syntax error - result from incorrectly constructed code
- Run time errors - detected as the program is running
- Logic error - when the code does not perform the way we indented

2. What are the two major categories of quality assurance testing?
- Error based testing
- Scenario based testing
3. Define error based testing?

Error based testing searches a given class's method for particular clues of interest then describe how these clues should be tested. Eg: to test the payroll computation method of an employee class:

```java
anEmployee.computePayroll(hours)
```

To test this method we must try different values for hours (say 100). To see if the program can handle them. This is also called testing the boundary conditions.

4. Define scenario based testing?

Scenario based testing also called usage based testing concentrates on what the user does not what the product does. This means capturing use cases and the task users perform then performing them and their variants as tests.

5. Define black box testing?

In black box testing a system is represented who's internal working are not available for inspection. The test item is treated as black since its logic is unknown. All that is known is the input and the output. Eg: user manual.

6. Define white box testing?

White box testing assumes that the specific logic is important and must be tested to guarantee the system's proper functioning. The main use of white box is in error based testing. In white box testing we are looking for bugs that have an low probability of execution.

7. Define path testing?

Its a form of white box testing which makes certain that each path in a objects method is executed at least once during testing. There are two types: Statement testing coverage and Branch testing coverage.

8. What is Statement testing coverage?

In statement testing coverage every statement in an objects method is tested by executing it at least once.

9. What is branch testing coverage?

In branch testing coverage tests are performed to ensure that every branch alternative has been executed at least once during some test.
10. What is Top-Down testing?

Top-down testing should find critical design errors early in the testing process and significantly improve the quality of the delivered software because of the iterative nature of the test.

11. What is Bottom Up testing?

In Bottom up testing each object is tested then its combined and their integration is tested and the messages passed among objects are tested by utilizing the top down approach. Bottom up testing leads to integration testing which leads to systems testing.

12. What is the impact of Object orientation on testing?

Some types of error could become less plausible (not worth testing for). Some types of errors could become more plausible (worth testing for now). Some new types of error might appear.

13. What is the objective of testing?

According to Myers
1. Testing is the process of executing a program with the intent of finding errors.
2. A good test case is the one that has a high probability of detecting an as-yet undiscovered error.
3. A successful test case is the one that detects an as-yet undiscovered error.

14. State few guidelines for developing quality assurance test cases?

1. Describe which feature or service your test attempts to cover.
2. If the test case is based on a use case it's a good idea to refer to the use case name
3. Test the normal use of objects methods
4. Test the abnormal but reasonable use of objects method.
5. Test the abnormal but unreasonable use of objects method.

15. What are the tests needed to create a test plan?

1. Objective of the test
2. Development of the test case
3. Test analysis

16. What is beta testing?

It's a popular inexpensive and effective way to test software on a select group of actual users of the system.
17. What is alpha testing?

In alpha testing, testing is done by in-house testers such as programmers, software engineers and internal users.

18. State a few guidelines for writing test plan developed by Thomas?

1. Requirements that dictate a specific appearance or format for a test plan
2. Test plan should contain a schedule and list of required resources
3. Determine what types of testing are necessary.
4. Configuration control system providing a way of tracking the changes to the code.
5. Keeping the plan up-to-date

19. What are the steps for successful testing?

1. Understand and communicate the business case for improved testing
2. Develop an internal infrastructure to support continuous testing
3. Look for leaders who will commit to and own the process
4. Measure and document your findings in a defect recording system.
5. Publicize improvements as they are made and let people know what they are doing better.

20. State Myers bug location and debugging principle?

Bug locating principles
1. Think
2. If you reach an impasse, sleep on it
3. If the impasse remains describe the problem to someone else
4. Use debugging tools
5. Experimentation should be done as a last resort

Debugging principles
1. Where there one bug, there is likely to be another
2. Fix the error not just the symptom of it
3. Probability of the solution being correct drops as the size of the program increases.
4. Beware of the possibility that an error correction will create a new error

UNIT V

1. What is a DB model? What are the types?

It is a collection of logical constructs used to represent the data structure and data relationships within the database. There are
- Hierarchal model
- Network model
- Relational model
2. What is client server computing?

   It is the logical extension of modular programming. The fundamental assumption is that separation of large piece of software into its constant parts which creates the possibility for easier development and better maintainability.

3. What is a middleware?

   It is the network operating system which is the key element of connectivity is known as middleware.

4. What are the components involved in client-server application?

   - User interface
   - Business processing
   - Dataprocessing

5. Define cooperative processing?

   It is the computing that requires two or more distinct processors to complete a single transaction. It is related to both distributed and client server processing.

6. Define CORBA?

   Common Object Request Broker Architecture a standard proposed as a means to integrate distributed heterogenous business application and data.

7. What are the two tasks performed by access layer?

   1. Translate the request
   2. Transulate the result

8. What is the idea behind access layer?

   It is to create a set of classes that know how to communicate with a datasource, regardless of data it must be able to translate data-related requests from business layer into appropriate protocol for data access.