

2 marks:

1. What is Model-View Separation Principle:

A Model-View Separation principle states that model (domain) objects should not have direct knowledge of view (presentation) objects, at least as view objects. Ex: a Register or Sale object should not directly send a message to a GUI window object ProcessSaleFrame.

2. Define Operation Contract with Syntax.

A UML operation contract identifies system state changes when an operation happens. Effectively, it will define what each system operation does.

Syntax:

Name: appropriateName

Responsibilities: Perform a function

Cross References: System functions and Use cases

Exceptions: none

Preconditions: something or some relationship exists.

Postconditions: An association was formed.

3. List the modifications made in logical architecture refinement.

Modifications made are:

- \* Inter layer and inter package coupling.
- \* Inter layer and inter package interaction scenarios.
- \* Collaborations with layers patterns.
- \* Session facades and the application layer.
- \* System operation and layers.

4. What are activities that a macro process includes in Booch methodology?

Macro Process:

- \* Conceptualization.
- \* Analysis
- \* Design
- \* Evolution.
- \* Maintenance.

5. For what purpose RDD is used?

RDD  $\Rightarrow$  Responsibility Driven Design.

These are used for.

1. Doing responsibilities of an object:
  - \* Doing something itself.
  - \* Initiating action in other objects.
2. Knowing responsibilities of an object:
  - \* Knowing about private encapsulated data.
  - \* Knowing about related objects.

6. Define high cohesion with example in grasp pattern.

High cohesion which includes defined purposes of classes, ability to reuse code. It is important to have code that is clean. Objects need to be manageable, easy to maintain. To have High cohesion, a class should have one job.

Ex: A game piece should move across the board. It should not need to setup the board or define moves for other players.

7. Define grasp and list its patterns.

GRASP - General Responsibility Assignment Software Principles help guide object-oriented design by clearly outlining WHO does WHAT: object or class is responsible for what action or job?

PATTERNS:

- \* Creator
- \* Information Expert
- \* Low coupling
- \* Controller
- \* High cohesion

8. Write down the elements used in composite pattern.

- \* Component
- \* Leaf
- \* Composite
- \* Client

9. Differentiate Forward engineering and reverse engineering.

## Forward engineering

- \* In this, applications are developed with the given requirements.
- \* It is a high proficiency skill.
- \* It is used to create new software applications from scratch.
- \* It takes more time to develop an application.

## Reverse engineering

In this, the information is collected from the given application.

It is a low proficiency skill.

It is used to modify and improve an existing software application.

It takes less time to develop an application.

## 10. Define TDD.

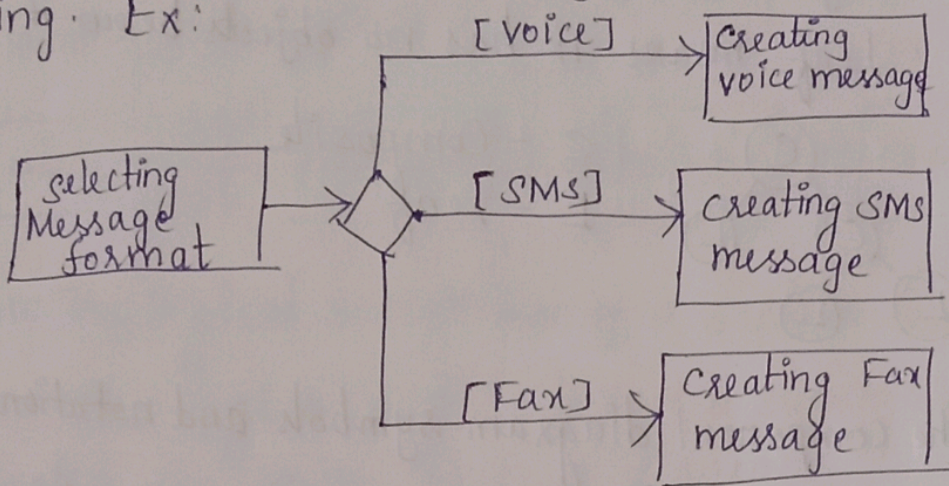
Test Driven Development (TDD) is the process in which test cases are written before the code that validates those cases. It depends on repetition of a very short development cycle. It is a technique in which automated unit tests are used.

## 11. What is an artifact?

- \* Artifacts are deployed on nodes and specify the physical pieces of information that the deployment and operation of a system uses or produces.
- \* Artifacts can be supported for deployment on several kinds of nodes.

12. Define choice pseudo ~~code~~ state in UML state machine diagram.

A choice pseudo-state is shown as a diamond with one transition arriving and two or more transitions leaving. Ex:



13. How the logical architecture of a system can be illustrated?

The logical architecture of a system can be illustrated by using Package diagrams.

A UML package diagram provides a way to group elements. It can group anything: classes, other packages, use cases and so on. Nesting packages is very common.

14. What are the design decisions made at architectural level in collaborations with layers pattern?

Two design decisions at an architectural level are:

1. What are the big parts?

2. How are they connected?

\* The architectural layer pattern guides defining the big parts.

\* Micro architectural design patterns such as Facade, Controller and observer are used for design of connections b/w layers and package.

15. What are the implementation methods used in OOSE

The Jacobson methodology, also known as Object-Oriented Software Engineering (OOSE) is a method used to plan, design and implement object-oriented software. The method is broken down into five parts: a set of requirements, an analysis, a design, an implementation and a testing model.

16. Define Information Expert in Grasp pattern.

\* Information expert is a principle used to determine where a delegate responsibilities. These responsibilities include methods, computed fields and so on.

\* Information expert pattern states that we need to assign responsibilities to the right expert.

17. What are the common ways through which visibility can be achieved between two objects?

\* Attribute visibility : B is an attribute of A.

\* Parameter visibility : B is a parameter of a method of A.

\* Local visibility : B is a (non-parameter) local object in a method of A.

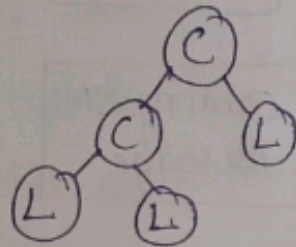
\* Global visibility : B is in some way globally visible.



18. What are the ways in which composite design pattern treats each node?

\* Composite : Composite means it can have other objects below it.

\* Leaf : Leaf means it has no objects below it.

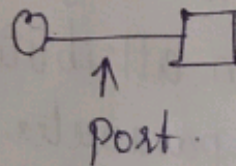
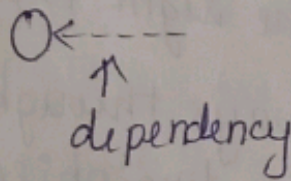
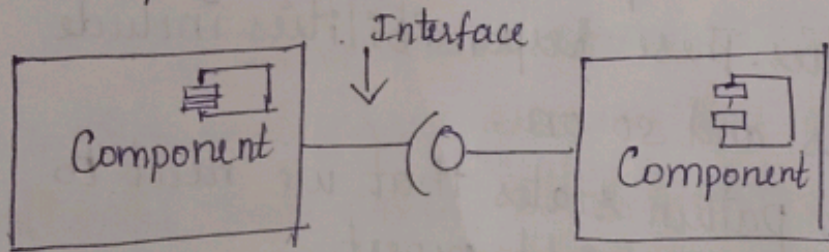


C - Composite  
L - Leaf.

19. Draw the component diagram symbols and notations.

\* Component      \* Interface

\* Dependencies      \* Port.



20. List the various UML tools which are currently used.

\* Star UML

\* Argo UML

\* Umbrello UML Modeller

\* Acceleo.

\* GenMy Model.

\* UML Tools List  
@wiki

21. What is OOT?

\* Object-oriented Testing is a collection of testing techniques to verify and validate object-oriented software.

\* Testing takes place.

