

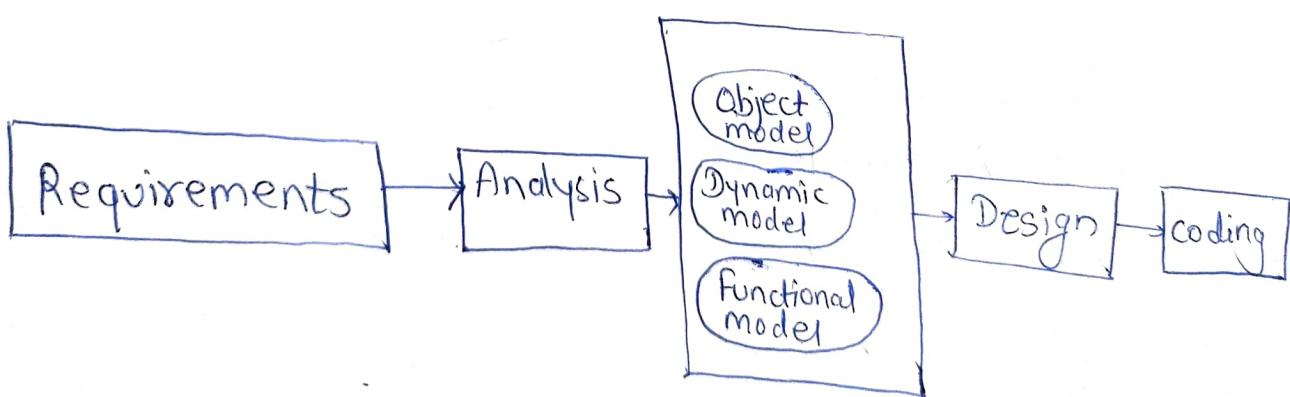
SMD

Software Modelling & Design

UNIT-I →

Q1] Explain Object Modelling Technique (OMT) by Rumbaugh / Phases of OMT

- • Object modelling technique is proposed by Rumbaugh
- This technique describes the method of analysing, design & implementation of system using ~~as OMT~~
 - It is fast technique of identifying objects.
 - It describes three models - object model, dynamic model & functional model
 - This technique describes three aspects of the system
 - Static, dynamic & functional behaviour of the system
 - It has four phase - Analysis, system design, object design & implementation



OMT has proposed three main types of models:

1) Object model: The object model represent the static modelling. Main concept are classes & associations with attribute & operation. Aggregation & Generalization are predefined relationship.

2]

Dynamic modelling → Dynamic modelling represent the state/transition view on the model. main concept are states, transition between states & event to be trigger transition. Generalization & Aggregation are predefined relationships.

3]

Functional Model → Functional model handles the process of model, by designing data flow diagrams. main concepts are process, dataflow, data store & factors.

4]

Explain Object oriented Analysis by Codd Yourdon

- • The focus of object oriented Analysis proposed by Codd Yourdon is on system analysis.
- This method is based on five major steps
These steps are also called as SOSAS
 - S → Subject
 - O → Object
 - S → Structure
 - A → Attributes
 - S → Services

The Unified Approach

Q Explain process & components of unified approach
→ ~~used~~ with neat diagram.

→ The unified approach consists of following processes,

1} Use - case driven development

2} Object-oriented analysis

3} Object-orient design.

4} Development

5} Testing.

2] Object-oriented ~~design~~ → Analysis is the process of extracting the needs of a system.

• The goal of object-oriented analysis is to first understand the domain of the problem & to understand how user use the system.

3] Object-oriented design → It plays major role in performance process.

Process →

1} Designing classes , attributes , method , association

2} Design access layer

3} Design prototype for user Interface

4} Development & testing

5} Refine the design.

Layered approach to Software development.

1) View layer →

- View layer consists of graphical user Interface object.
- Once the analysis is complete we can start designing the user interface object.
- The main goal of a user interface is to display and obtain required information in accessible and efficient manner.
- The ~~the~~ good UI can help in user satisfaction.

2) Business layer →

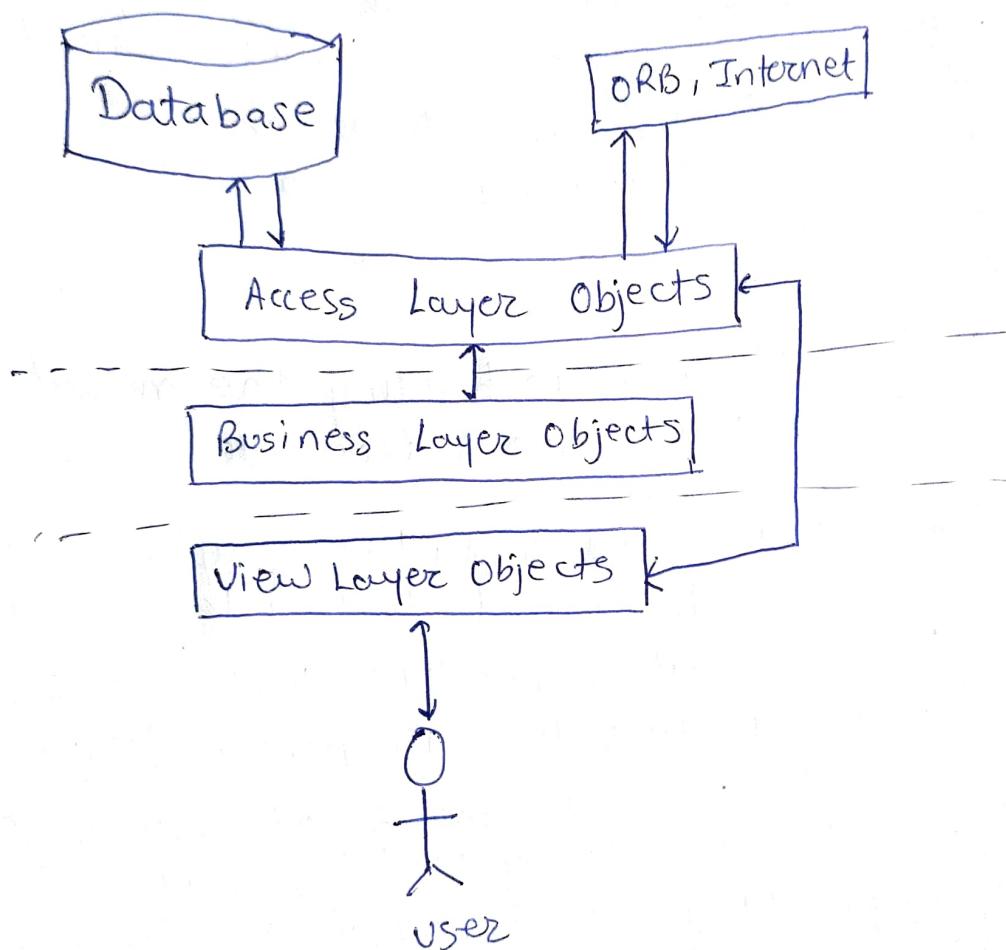
- The business layer contains all the objects that represent the business.
- This is where the real objects such as order, summary, customer, student, and so on.
- It captures static & dynamic relationship between collection of business objects.
- Static relationship includes object association & aggregation.
- Dynamic relationship shows how the business objects interact to perform tasks.

3) Access Layer →

- It is responsible to communicate with the place where the data actually reside.
- The primary goal of access layer is to provide link between business or view object and data storage.

Major tasks of access layer

- 1] Translate a request → [Business layer → appropriate protocol for data access]
- 2] Translate the result → vice versa



Q Write a short note on UML : A Language

- The UML is language for
- 1) Visualizing
 - 2) Specifying
 - 3) Constructing
 - 4) Documenting

1] Visualizing → UML contains a collection of graphical symbols. For each symbol there is well defined semantic. Due to this one developer can interpret the model without any ambiguity.

2] ~~Specifying~~

2] Specification → UML helps in building the model that are precise, unambiguous & complete.

3] Construction → UML is not simply the graphical tool but one can map the model directly to programming language. It means it is possible to map the diagrams drawn in UML to a programming language such as Java, C++ etc.. The process of mapping from model to programming model is called forward engineering & from programming to UML model is called reverse engineering.

4] Documentation :

- 1] Requirements
- 2] Design
- 3] Implementation
- 4] Project planning
- 5] Testing
- 6] Maintenance.

Pros :

- UML is easy to use
- Easy to understand
- Self explanatory
- Programming Independent

Cons

- Time consuming
- Not possible to showcase all details

Q] Explain the Basic Building blocks of UML.
OR
→ Explain the components of UML.

→ Basic Building Blocks.

- 1] Things
- 2] Relationship
- 3] Diagram.

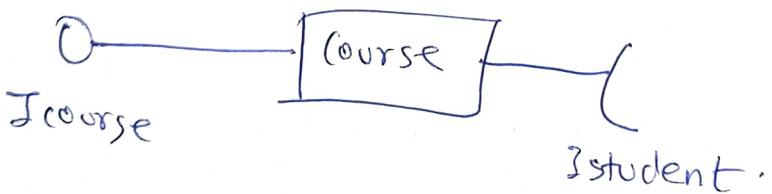
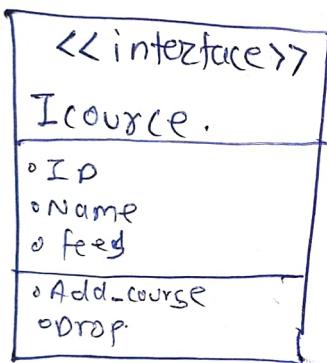
1] Things → There are four kind of Things

- 1] Structural Thing.
- 2] Behavioural Thing.
- 3] Grouping Thing.
- 4] Annotational Thing.

I] Structural Things: This are the static part of the model. It may contain conceptual or physical elements. Structural Things are also called as classifiers.

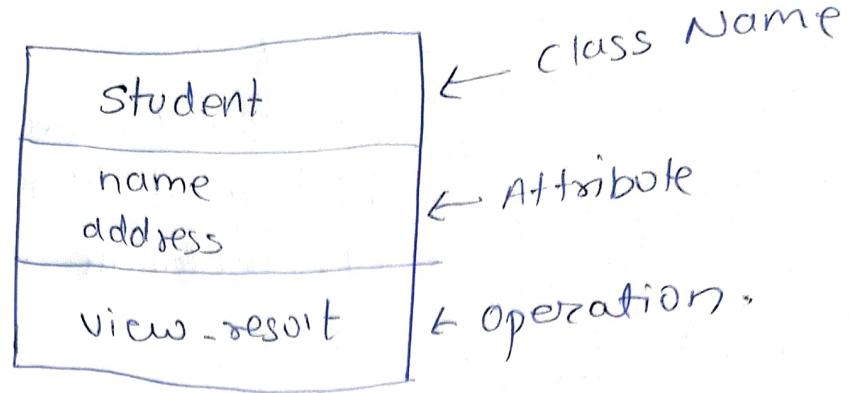
a] Interface → It is collection operations that specify the service of class or component. Interface describes the externally visible behaviour of class.

◦ It might describe the complete behaviour or part of behaviour & it is denoted with keyword `<<interface>>` at above the name.

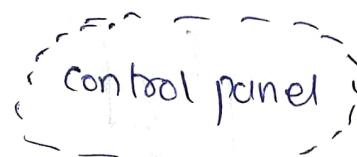


b] Collaboration:

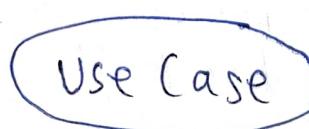
b] Class → A class is set of objects that share the same attributes, operations, relationships. It is represented using rectangle, with three sections • Class Name, Attributes & operation.



3] Collaboration → It represents the interaction with other element which provides cooperative behaviour. There can be multiple collaborations. It is represented by dotted elliptical shape.



4] Use case → Use case represents by specific scenario. by meant of set of actions that a system can perform. It tells how the system shall work. It gives the overview of system. and actions are driven by the actor. It is represented by solid elliptical shape.

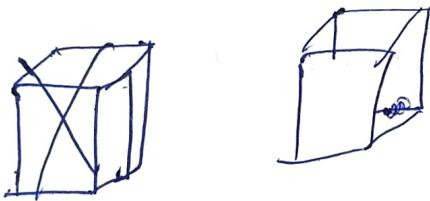


5] Active class → An active class is class whose objects own one or more process or threads. It is similar to class except its object represent element whose behaviour is concurrent with other element. It shows all things in parallel.

6] Component → It represents the software module with well define structure. It represent the physical entities or logical elements. and it is represented by rectangle with tabs.



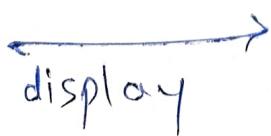
7] Node → It is a deployment diagram. It is an important block where physical fact of software is noted. It is represented using 3D box



2] Behaviour Thing

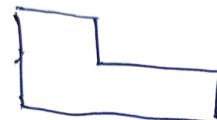
- It is dynamic part of the model. It represent the behavior of the system.

1] Interaction → The message exchange among the set of objects to perform certain task. It is represented by direct line.

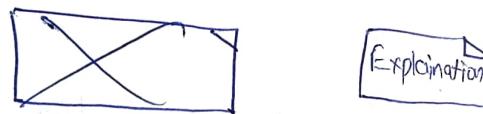


2] State machine → Shows the events that are to be perform. It shows the actions that are performing.

- 3] Grouping Thing → • The primary ~~king~~ kind of grouping is called package.
• package is the mechanism of organizing the design itself. package can group anything like structural things, behavioural things and so on.
• It is represented as follow



- 4] Annotational things → • It is for explanation purpose
• It works as note.
• Represented by as follow



2] Relationship

- 1] Association → It is structural relationship among two classes and it is static in nature. It shows fixed relationship.
• Represented by arrow. (→)

- 2] Aggregation → It is extraordinary form of relationship
It is represented by diamond symbol (◇)

3] Composition → It is similar to aggregation. difference is that it is shown with filled diamond ◆

4] Generalization (→) It is kind of relationship between superclass & subclass.

5] Dependancy (--->) It shows the relationship between two or more element. generally, the arrow gives the direction of class is depended on.

3] Diagram in UML

1] Static Diagram

1) Class → set of classes, interface collaboration & their relationship

2) Object → Object are instance of the class & these diagram shows the relationship among the different objects.

3) Use Case → Shows how the system will look like., gives the overview of system.

2] Dynamic Diagram

1) Sequence Diagram → modelling the behaviour of the system

2) State Diagram → Represents the state, transition, event.

3) Activity Diagram → Represent the functionality, shows the flow.

3] Implementation Diagram

1) Component → collection of components. represents the internal structure of the system

2) Deployment → Run time processing of component is represented by these diagram.

Q Explain common mechanism in UML.

→ 1] Specifications:

- Along with graphics the UML provides textual statements that have some special semantic & syntax.
- By using specification, UML ensures that the same element can be used across different diagram without redundancy.
- It is possible to build the system gradually by adding the diagram one by one and by adding semantics to these diagrams.

2] Adornments:

- Adornments means Additional notation which provides more details. This helps in proper visualization of the system.
- For example → Along with the ~~system~~ rectangular representation we can specify the attributes and operations of the class in details. We can specify if the class is public or private. A adornments is called

3] Common divisions

1] Classes & Objects

Class → It is the blueprint for the objects.

It defines the attributes & methods that the class will have.

for example Employee can be class & sachin, & sachin can be object

Object → It is an instance of a class. It represents a real-world entity.

2] Interface & implementation

- An interface defines the rule of interaction
- Implementation is the concrete realization of the rules defined in the interface.

4] Extensibility mechanism

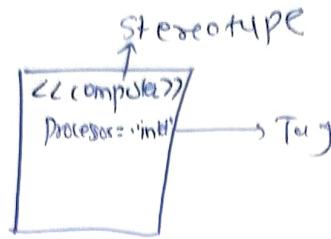
It includes

1] ~~Stereotype~~
1] ~~Stereotype~~

2] Tagged value 3] Constraints

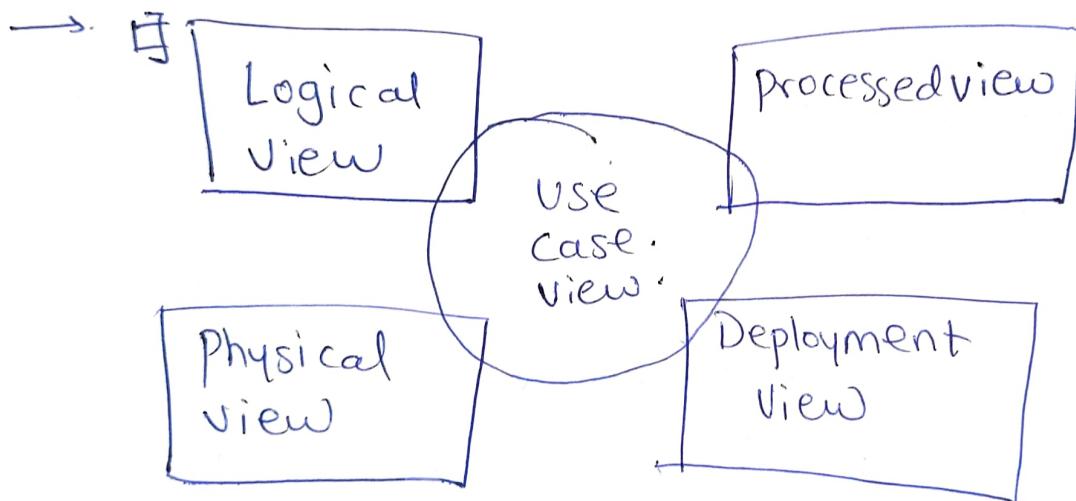
1] Stereotype → Used to extend the UML notation elements. A stereotype provides the capability to create new kind of modeling element.

2] Tagged Values : Stereotype declared the set of tags properties of stereotype used is called tags.



3] Constraints : Used to define condition or rules on UML element. It is the boolean value True or False.

Q] Explain 4+1 view architecture.



- 1] Logical view → • It is the design view & is responsible for defining the software system
- Focuses on System functionality as seen by user
 - Useful for designers & developers to understand system ~~de~~ behavior.

Diagram used

- 1] Class Diagram
- 2] Object Diagram

2] process view

- It includes process and threads involved in the software system
- Process view mainly refers to the scalability, performance, output & fault tolerance of the system
- Useful for architect & designers.

Diagram used

- 1] ~~Component~~ Activity Diagram
- 2] State diagram

3] physical View / Deployment view .

- It describes the component such as hardware required for deployment.
- Focuses on network topology & communication

Diagram Used

- Deployment Diagram,

4] Development view / Implementation View

- focuses on the internal organization of the system
- Represents Modules, packages, components.

Diagram used

- Component
- Package.

5] Use case View (+I view)

- Shows how the user will interact with system
- Overview of system design
- captures the requirements
- Helps stakeholders & business analyst.

Diagram Used

- Use case diagram.