

- Q. UML :- UML is a specification language which is used for visualization and creating an model of final product.
- Q. UML has been used as a specification language or a general purpose modelling language.

Types of UML Diagram :-

UML Diagram

Structure Diagram

- Class Diagram
- Component Diagram
- Composite Structure Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram

Behavior Diagram

- Activity Diagram
- Use Case Diagram
- State Machine Diagram
- Interaction Diagram

①. Structure Diagram show the static aspect of modelling

②. These diagram include composite structure diagram, deployment diagram, package, profile, class & object diagram and component diagram

Class Diagram

| |
|-------------|
| Object Name |
| Attribute |
| Operations |

UML

① Object Diagram :- The Object UML diagram also known as object oriented diagram is used to represent the static view of a system and portray ~~an~~ instance of an UML class diagram.

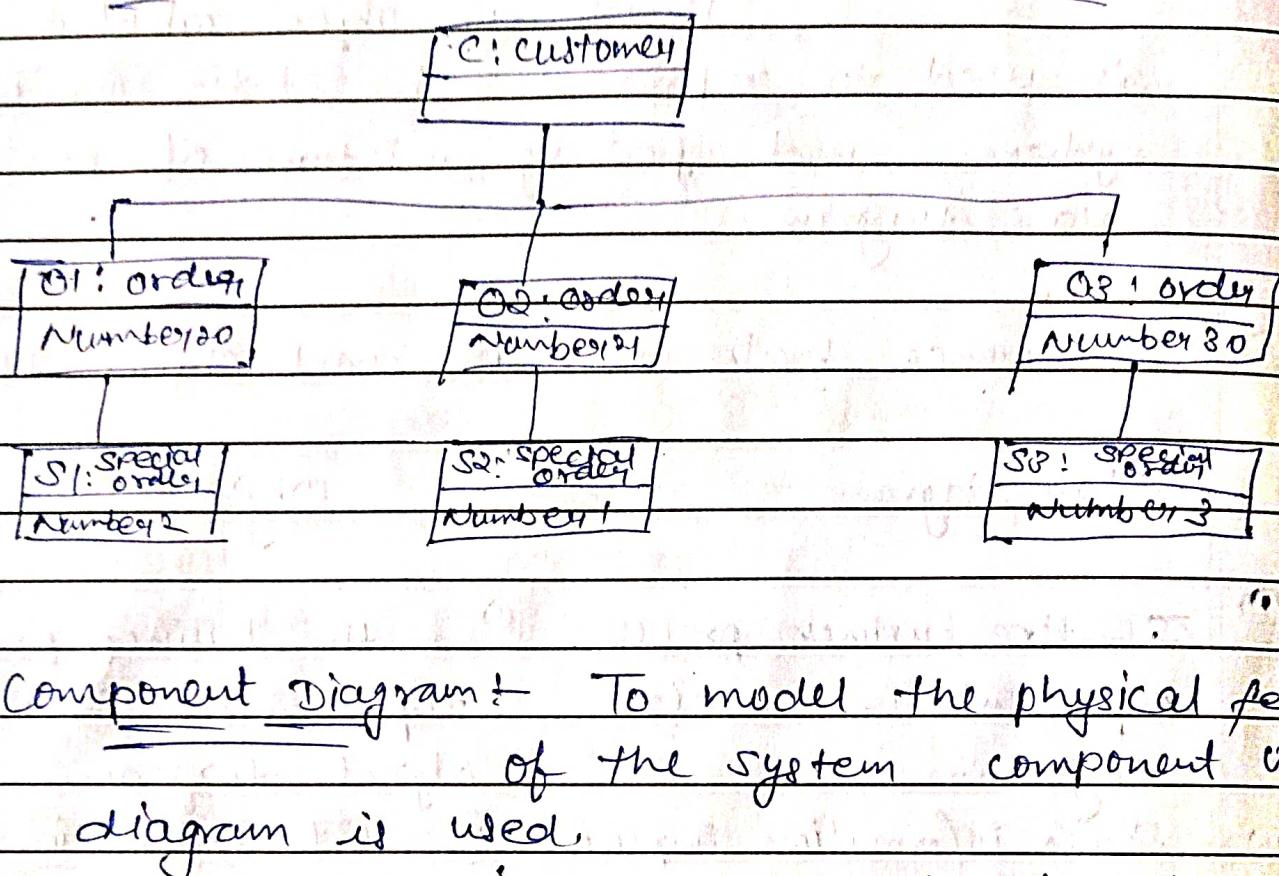
② Difference between Class and Object Diagram

Class DiagramObject Diagram

| | | |
|---|---|--|
| ① | Entire business process represented in one class diagram. | ① Entire business process represented in no of object diagram. |
| ② | State information implicit present. | ② State information explicitly present. |
| ③ | Implicit gives information for composit structure. | ③ Explicit gives information for object structure. |
| ④ | Sub class present once in class diagram. | ④ Same sub class can represent in many object diagram. |
| ⑤ | Visibility - Private, public, protected | ⑤ Visibility - Public, protected. |

⑫ Object Diagram +

Order Management System



⑬ Component Diagram + To model the physical features of the system component UML diagram is used.

Physical aspect include elements like libraries, executable files, documents, etc. which are placed in a node.

Component diagram are used to describe the component utilize to execute a functionality and not to describe the functionality as a whole.

There are three element in UML document diagram:

- ① Component
- ② Interface
- ③ Dependency.

Date :
Page :

Component

C1

interface Component

C2

(Basic of component Diagram)

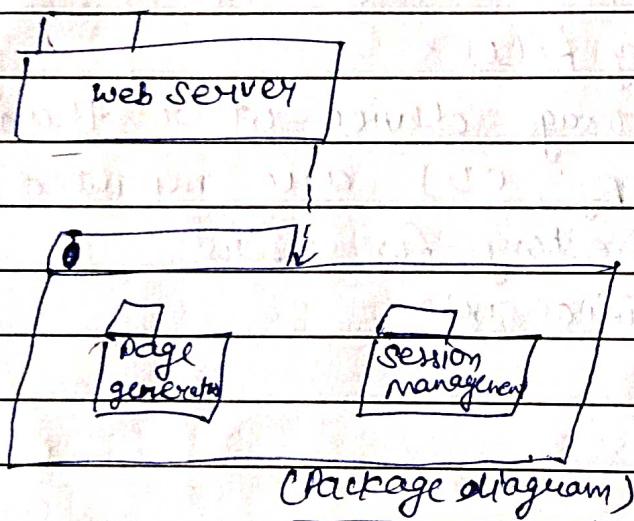
(i) Package Diagram :- In Package Diagram model elements are organised in various ~~and~~ ^{middle} to large scale project.

(ii) A package in a package diagram is basically a collection of different logically associated UML element.

(iii) These packages are represented as five folders which we can use in different UML diagram.

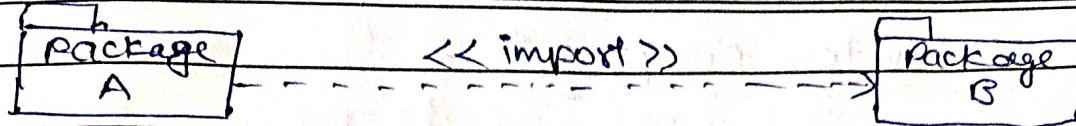
(iv) There are two sub types of dependencies involved in a package diagram:

(a) <<Import>> (b) <<Access>>

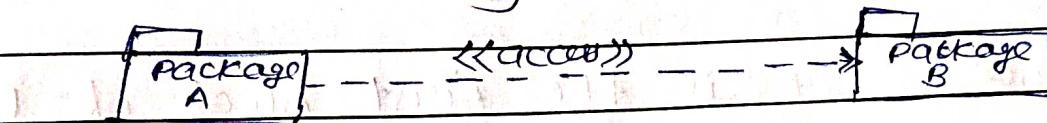


(v) import command → By using this command one package can import the functionalities of another package.

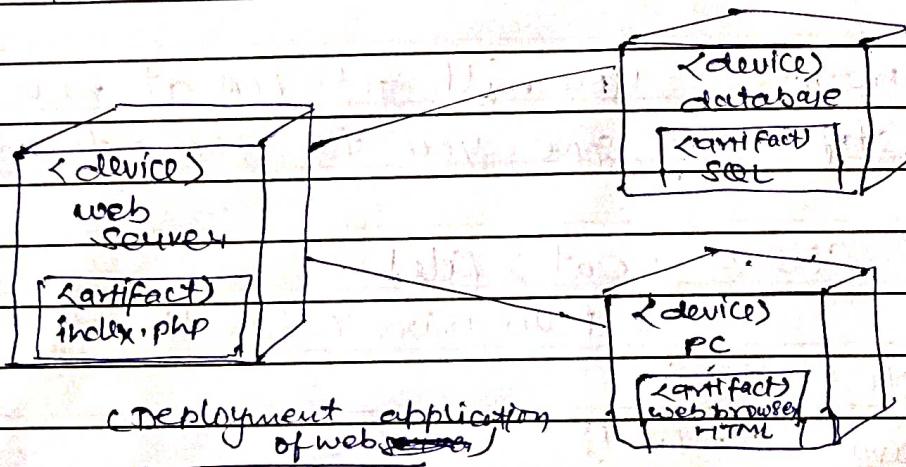
(vi) The imported element gets added to the namespace.
Import is a public command.



- ⑨ `(access)` command : It is used, when one package needs functionalities of the other package.
- ⑩ ~~access~~ ^{command} basically a private form of import -



- ⑨ Deployment Diagram :- Deployment diagram is representing the hardware based upon which the software is going to the for performing the functionality.
- ⑩ Deployment diagram shows the interaction between SW and HW to complete the function.



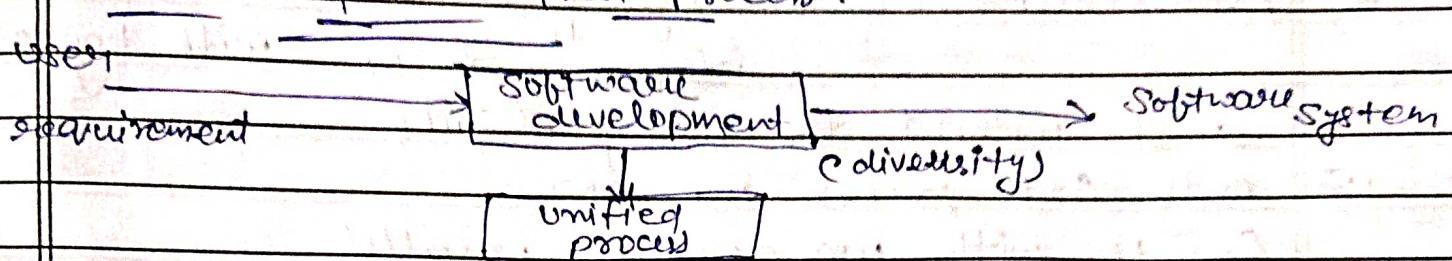
- ⑪ The 3d boxes represent the basic hardware and software component. Lines represent the relationship between nodes and the small shape.

Unified Process :-

- ① what does Unified Process do?
- Provide guidance to the other team activities.
 - Integrate team work and individual work.
 - Specify artifacts
 - Offer criteria for monitoring and measuring.
- ② Reason for unified process

- (2) Software become more complex and is updated fast.
- (3) Software developer uses method that are to hold
- (4) Development process is diverse.

Precursors for Unified process:



Key aspects of Unified Process:

- (1) Use case driven
- (2) Architecture Centric
- (3) Iterative and incremental

UML

- (1) Use case driven : Development process proceed through a series of workflow that derived from use cases.

Users - Someone or something that interact with system.

Use case - Interaction between user & system - what is the system supposed to

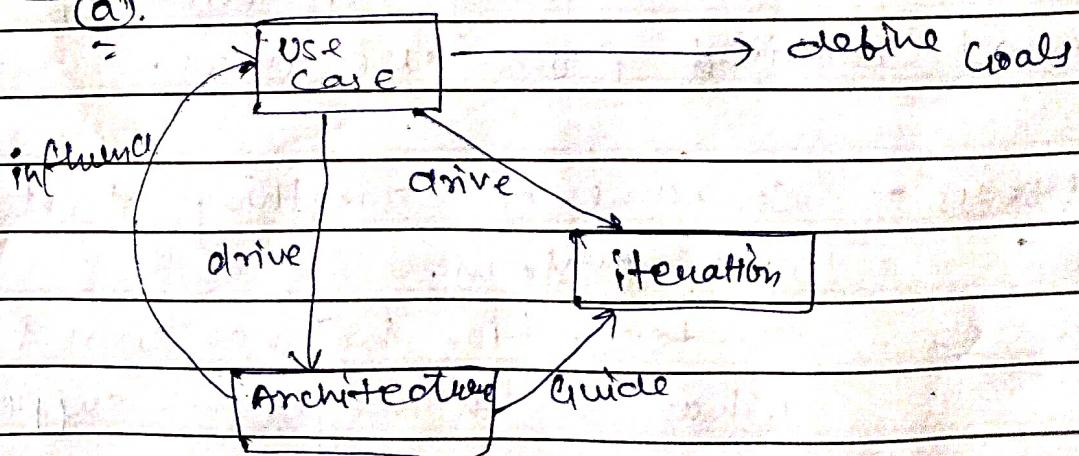
- for each user.
- ②. Initiate and Bind : Tool for specifying requirement
①. driving design.
- ⑩. Source of testing
- ⑪. Architecture Centric : It is the view of the whole ~~board~~ design with key characteristics and without too many details
only 5-10 ~~process~~ use cases
Growth with use cases in parallel.

⑫. Iterative & Incremental

- ①. Iteration : Steps in workflow.
 - a. Create a design for relevant use case.
 - b. Implement with component
 - c. Required iteration in logical order for economy.

⑬. Increments : Growth in the product

(a).



④ Use case Diagram :- First it show a system or application then it show the people organization or other system that interact with it and finally it show a basic flow of what the system or application does.

⑤ It is a very high level diagram and typically want show a lot of it is great way to communicate complex idea in a fairly basic way.



→ Actor or Roleplay (System)



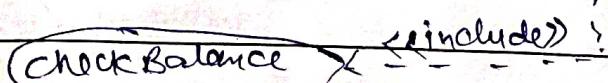
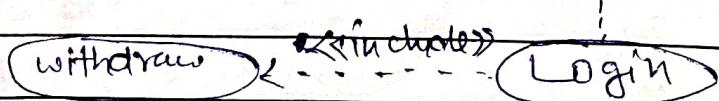
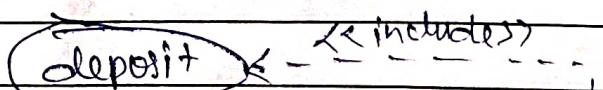
→ Use case (Capability)

— Connector

— → Generalization

- - - - - → Stereotype (Relationship)

⑥ Relationship :- (1) <-- includes --> implicit function



(11).

<<extend>> Explicit function

cart<<extend>>checkout

(2). Shopping application with the help of use case diagram \Rightarrow

