OOAD

M.Madhuri
19KQ5A0506
CSE-B

Assignment No. 04 Unit No. 04 Date : 28/12/21

1) Explain about the common mechanism in UML?

A) Common Mechanisms:-

It can be classified into:

i) Specification
ii) Adomments
iii) Common divisions
iv) Extensibility mechanisms.

i) Specification:-

Specification provides a textual statement describing interesting aspects of a system.

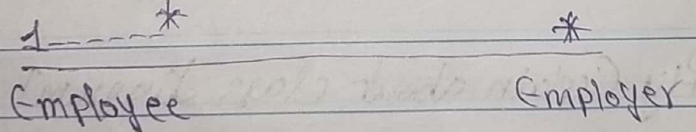
ii) Adomments:-

* Textual / graphical items added to the basic notation of an Element.

* They are used to explicit visual representation of those aspects of an element that are beyond that most important.

Example:-

The basic notation of association is line, but this cloud can be adorned with additional details, such as the role names and multiplicity of Each End.



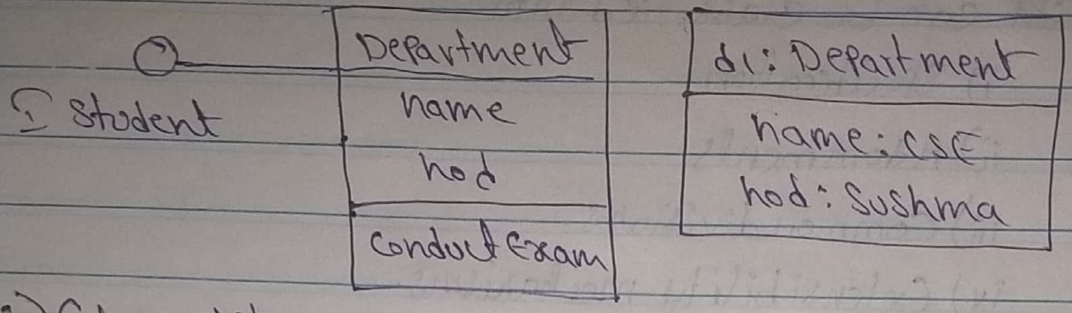
* Similarly, a class notation may highlight most important aspects of a class i.e, name, attributes and operations.

iii) Common Divisions:-

* In modeling object-oriented systems get divided in multiple ways.

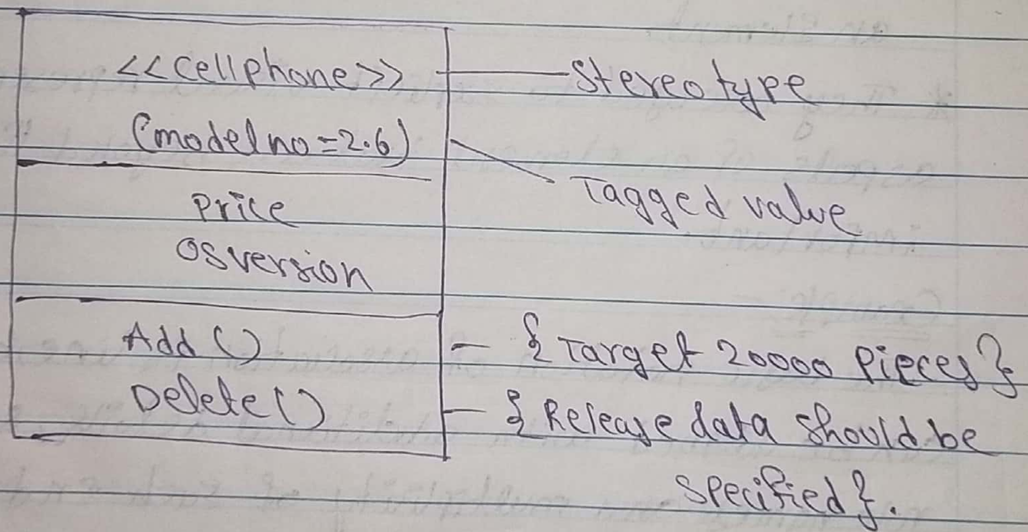
* for Example, class vs object, interface Vs implementation

* An object uses the same symbol as its class with its name underlined.



iv) Extensibility mechanisms:-

* Extensibility mechanisms allow extending the language in controlled ways.
* They include stereotypes, Tagged values and constraints.



2) Briefly Explain about class diagram?

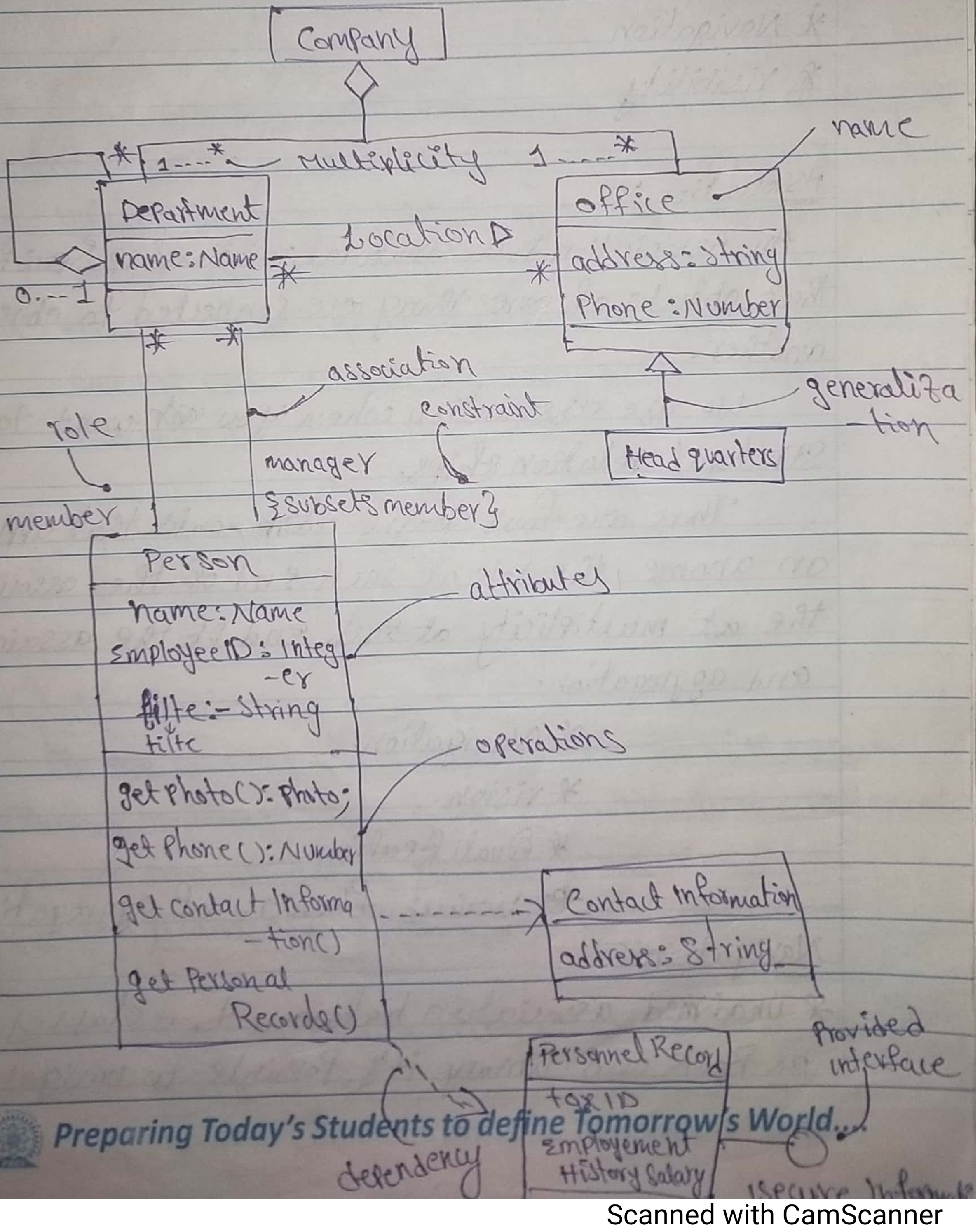
A) Class Diagram:-

* Class diagrams are the most common diagram found in modeling object-oriented system.
* A class diagram shows a set of classes, interfaces and collaborations and their relationships.
* Class diagrams are important not only for visualizing

Scanned with CamScanner

Specifying and documenting structural models, but also for constructing executable systems through forward and reverse engineering.

* Building software has much the same characteristics ~~exe~~ except that given the fluidity of software, we have the ability to define your own basic blocks from scratch.



* With the UML, you use class diagrams to visualize the static aspects of these building blocks and their relationships and to specify their details for construction, as you can see in figure.

3) Explain about advanced relationship in the structural model?

A) <u>Advanced relationship:-</u>

* Association
* Navigation
* Visibility

<u>Association:-</u>

An Association is a structural relationship specifying that objects of one thing are connected to objects of another.

We use association when you ~~ask~~ want to show structural relationships.

There are four basic adornments that apply to an aname, the role at each end of the association the ~~at~~ multiplicity at each end of the assciation and aggregation.

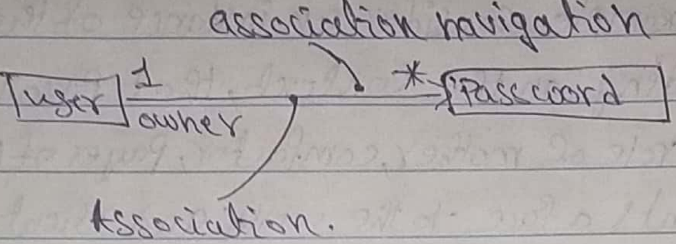
* Navigation.
* vision.
* Qualification.
* Various flavours of aggregation.

<u>Navigation:-</u>

* unadorned association between two classes, such as Book and library it's possible to navigate from

objects of one kind to objects of other kind.

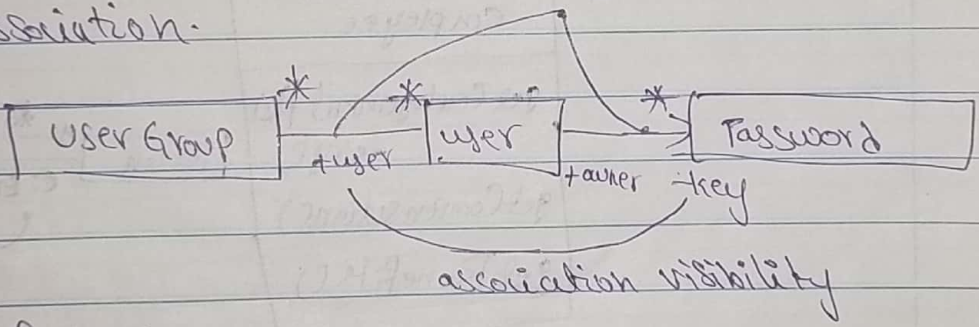
* However, there are some circumstances in which you'll want to limit navigation to just one direction.



Visibility:-

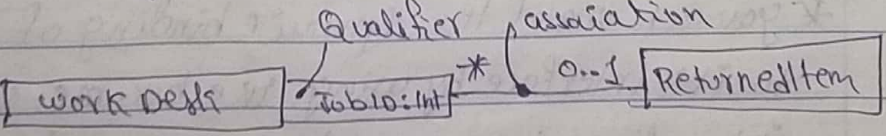
* Given an association between two classes, objects of one class can see and navigate to objects of the other unless otherwise restricted by an explicit statement of navigation.

* Private visibility indicates that objects at that end are not accessible to any objects outside the association.



Qualification:-

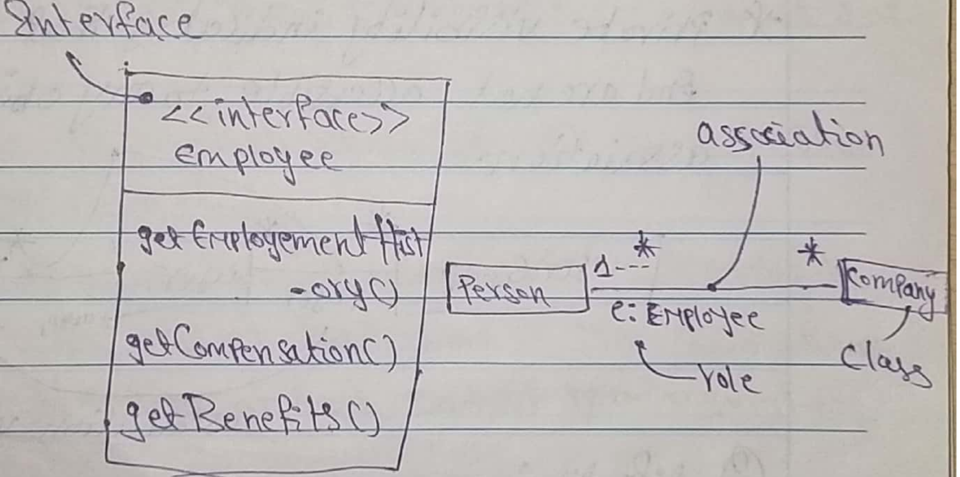
In the context of an association, one of the most common modeling idioms you'll encounter is the problem of look up.



4) what are the types and rules of structural modeling?

A) Types and roles:-

* A role names a behaviour of an entity participating in a Particular context
* Stated another way. a role is the face that an abstraction Presents to the world.
* for Example, Consider an instance of the class Person.
* Depending on the context, that Person instance may play the role of mother, comfactor, Payer of Bills, Employee Presents a face to the world and client that interact with it Expect a behaviour depending on the role that it Plays at the time.
* An instance of Person in the role of manager would Present a different set of properties than if the instance were Playing the role of Mother.



* A class diagram ~~[illegible]~~ like this one is useful for modeling the static binding of an abstraction to its interface
* you can model the dynamic binding of an abstraction to its interface by using the become stereotype in an interaction diagram, showing an object changing from one role to another.
* If you want to formally model the semantics of an.

Assignment No. Unit No. Date :

abstraction and its conformance to a specific interface you'll want to use the defined stereotype.

* Type is a stereotype of class, and you use it ~~specify a~~ Specify a domain of objects, together with the operations applicable to the objects of that type.
* The concept of type is ~~class~~ closely related to that of interface, except that a types definition may include attributes while an interface may not.