

UNIT - 1

INTRODUCTION TO MODELLING.

Q1) What is object orientation? what are its characteristics?

- Object orientation means organization of software as a collection of discrete objects.
- Each object consists of data structure & behaviour which are closely coupled.
- Following are the characteristics of OO:

1) Identity

- The identity refers to data arranged in distinguishable entities called objects.
- Each object is treated as inherent entity.
- Each object is distinct even if their attribute values are same.

2) Classification

- Classification is a technique in which objects of same data structure (attribute) & behaviour (operations) are grouped into class.
- Each class can define infinite no. of objects.
- Object is an instance of a class.
- Each object can have its own value for attributes & operations but these objects share the attributes & operations of other objects too.
- Eg. Shape can be a class & rectangle, triangle & square can be the objects of class shape.

3) Inheritance

- Inheritance allows the class to share property of another class in hierarchical manner.
- The superclass is a class whose properties are shared by the subclass.
- The subclass not only shares the properties of superclass but can add its own unique properties.

- Super class → general class
 - Subclass → specific class
 - Eg - Animal is a super class but Dog, Cat, Cow are sub classes.
 - Each subclass has its own unique feature.
 - Inheritance provides the property of reusability of code.
- 4) Polymorphism
- Poly means many & morphs means forms.
 - Polymorphism is a technique in which same operation can be defined differently for different classes.
 - Eg - operation clean can be used to clean the dish object, a cloth object or vegetable object.

Q2) What is OO Development? what are the stages of OO methodology?

- . In OO Development the application concepts are identified organized logically.
 - . The OO development & modelling is based performed by following
- i) Modelling concepts & not implementation
 - ii) OO Methodology
 - iii) Three Models usage
- Following are the steps of OO Methodology:
- a) Systems conception

In this phase requirements of the system can be specified by the business analysts or end users who want to use the system.

b) Analysis

- In this phase analyst scrutinizes & reframes requirement specified in the first system conception stage.
- He prepares an analysis model.

- This model is concise & precise abstraction of what the system must do.
- It should not specify how the system should do.
- Implementation decisions should not be taken at this stage.
- Analysis has two parts
 - Domain model
 - Application model.
- Real world objects are described in domain model.
- description of application model in parts of application - application model.

c) System design

- During system design, system architecture is built.
- Policies for the detailed design are decided.
- Tentative resource allocation is done in this phase.

d) Class design

- Class designers add details to the model using system design strategies.
- The domain & application models are elaborated using OO concepts.
- Focus of class design is on data structures & algorithms.
- Classes & relationships are developed in this phase.

e) Implementation

- In this phase, classes & relationships are transformed into the particular programming languages, databases & so on.
- The code must be flexible & extensive.

Q4. Describe Explain the three models used in OO technology.

→ There are three types of models which describe the system:

1) Class model.

- The basic static structure of the objects & the relationships are described in this model.
- The class diagram is a graphical representation of in which

the nodes represent classes & arc represent the relationships among these classes.

2) State model.

- The object changes its state over time.
- This can be represented using the state model.
- State diagram is a graphical notation in which the nodes represent states & the arcs represent the transitions between states caused by events.

3) Interaction model.

- The interaction model describes how objects interact with each other to perform some task.
- The use case diagram, sequence diagram & activity diagram are three kinds of diagrams drawn in the interaction model.
- use-case → represents functionality of the system
- sequence → represents interacting objects & time sequence of interaction
- activity → represents important activities performed by the system.

Q5. Explain OO Themes

→ i) Abstraction

- Abstraction is a mechanism in which only essential aspects of an application are focused while others are ignored.
- Abstraction means examining selective area of the problem.
- The goal of abstraction is to separate out all important aspects of the specific area of problem & ignore all unimportant aspects of that area.

ii) Encapsulation

- Encapsulation is also known as information hiding.
- In this technique the internal implementation is hidden from other object.
- In encapsulation, the data structure & behaviour are bound

together in one entity class.

- Due to encapsulation one can make changes in the behaviour of one object without affecting remaining part of the system.

iii) Sharing

- In OO technique sharing of code at different levels is possible.
- It can be done with the help of inheritance.
- The code gets reduced & concept clarity can be brought in the application development.
- Sharing is possible not only in between modules but also in future projects.

iv) Synergy

- The characteristics of OO → identity, classification, inheritance & polymorphism can be used in isolation.
- But if all the properties are used together, they complement each other synergistically.
- This is beneficial for OO approach of development.

Q6. Why are models created?

→ 1) Communication with customers

Architects & system designers represent the model to their customers in order to give a mock representation of the system.

2) Testing of physical entity

If the actual system is built & then tested then it is costly affair.

The simulation of model is not only cheaper but informs developers about the problems that may occur in actual system.

3) Visualization

- Using the model system can be represented for visualization.
- The customer can visualize the system due to model.

(Q7) What is a model? Explain different types of models. What is the relationship between them?

- Model is a way to create distinct views of the system.
The models evolve during development.
They have two dimensions : first dimension includes views such class model, state model & interaction model.
The second dimension includes analysis, design & implementation.
There are three types of models:

1) Class Model

- Class model describes the structure of the object, the relationships of one object with another object, attributes & operations of the objects.
- The state & interaction models use the context of the class model.
- The goal of constructing class model is to capture the concepts from real world that are useful for application.
- The class model is represented by the class diagram.
- The classes in this diagram define the attributes & operations of each object.
- The association relationship is used to link the classes together.
- The generalization relationship is used for a group of classes to share common data structures & operations.

2) State Model

- State model is concerned with time & sequencing of objects operations of the objects.
- On occurrences of events the object state changes their current state.
- The state model captures control aspect of the system.
- The state model is represented by state diagrams.
- Each state diagram becomes operations of ~~an~~ object in the class model.
- References between the state diagram become interactions in

interaction model.

3) Interaction model

- Interaction model describe how one object collaborates with other in order to achieve behaviour of the system.
- The overall behaviour of the system can be represented with the help of state & interaction diagram.
- Interaction diagram includes use case diagram, sequence diagram & activity diagram.
- The use case diagram shows how outsiders too interact with the system to achieve / gain functionality.
- The sequence diagram shows how objects interact & time sequence of interaction.
- The activity diagram shows flow of control among objects.

* Relationship among models

- The class model describes data structures.
- The state & interaction model represent the operations performed on these data structures.
- In class model every class performs some set of operations.
- These operations cause the events & actions which are highlighted by state & interaction model.

Q8) Objects

- The main purpose of class model is to describe objects.
- Object is an instance or occurrence of a class.
- Objects can be conceptual identities, real-world entities or important things from implementation POV.
- Objects are distinguishable entities.
- Identity means objects are distinct due to their inherent existence & not by their description.

* Classes.

- Class is a group of objects having same attributes & operations, relationships & semantic.
- Objects of particular class have same attributes & behaviour but each object is different from the other due to values of its attribute, different behaviour & relationship with other object.
- Objects in a class share a common semantic purpose.
- Each object belongs to some class & that object is aware of its belonging class.
- Grouping the object into corresponding classes make the design abstract.

* Class Diagrams.

- Class diagram is a graphical representation used for modelling, classes & their relationships.
- It describes all possible objects belonging to the classes.
- It is used for abstract modelling & implementing actual program.
- Class diagram is concise & can be understood easily.

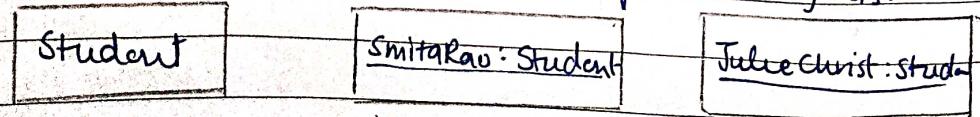
* Object Diagram.

- The object diagram represents the objects & their relationships.
- Using one class diagram, various object diagrams can be created.
- Object diagram is used for documenting test cases.

• Class & object is represented graphically by a box.

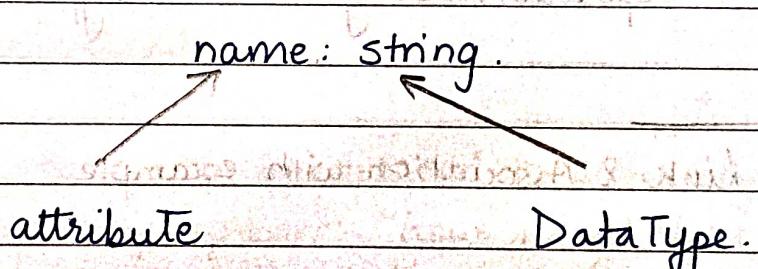
• Class name must be written at the centre of the box & first letter is capitalised.

• Object name is followed by a colon & class name. Everything must be underlined in the box that represents objects.



* Values & Attributes

- Attribute is named property of a class & it holds some value.
- Different objects may have same or different values for its attributes.
- The name of attribute is often always unique within the class.
- The values are always assigned to the attributes.
- Values alone have no identity.
- They can be recognized with the help of belonging attributes.
- Attributes are always written in the second compartment of the class.



* Operations & Methods

- Objects have procedures or functions called methods.
- All the objects in the same class have same set of operations.
- The same operations can be applied to different classes or it can be used for different purposes. They are known as polymorphic operations.
- Method is an implementation of operation for a class.
- Some piece of code is used to implement each method.
- Any no. of arguments can be passed to the operations.
- Generic word used for attribute or operation is called feature.
- The operations are always written in the third compartment of the class box.

* Summary of UML Notation.

Class Name.

attribute 1 : datatype = Def Value

attribute n : datatype = Def Value

operation (arg list) : returnType

operations (arg list) : returnType

Q8. Explain Link & Association with example.



Link is a connection between the objects.

It represents a simple association between two or one object with another.

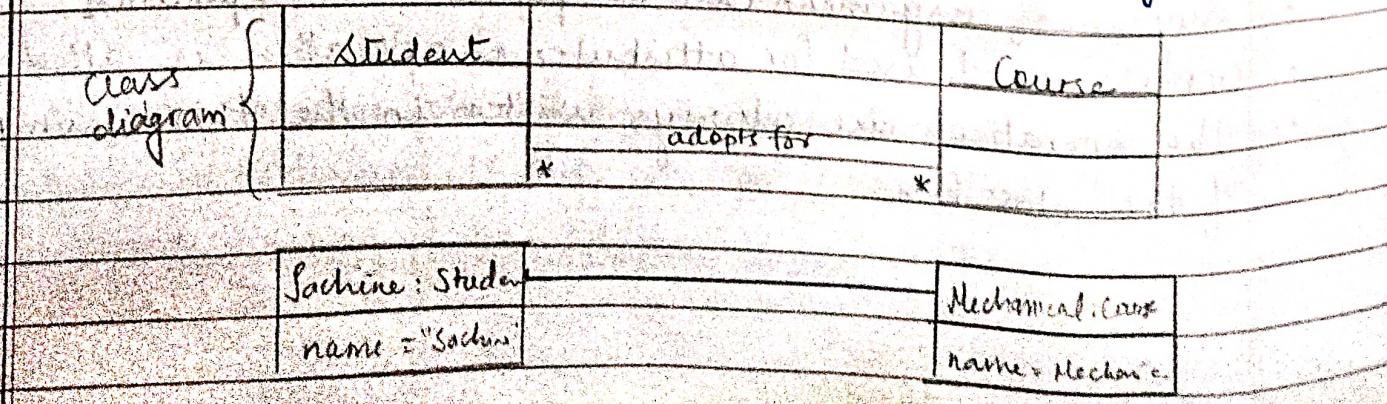
Mathematically link can be represented as a list of object.

A link is basically instance of an object association.

An association is a group of links that have common structure & common semantics.

The link between two objects & association is between two classes.

Eg - Following is a part of Online Course Management system



- If the model is unambiguous then association name is optional.
- Associations are inherently bidirectional, but are read in a particular direction.
Eg: In the class diagram, the student adopts for particular course represents the flow in one direction.
- Associations can be implemented with the help of references.
- Reference means the attribute of one object can be referred by another associated object.
- Due to encapsulation the operations are kept private to a class.
- But the encapsulation association breaks this encapsulation.

Q9. Explain Multiplicity.

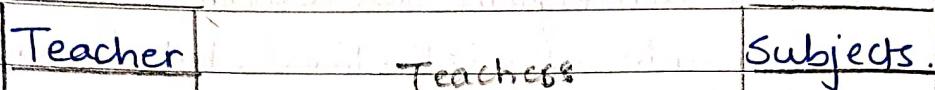
-
- Multiple objects can be related to some objects.
 - The multiplicity represents "how many" objects are connected.
 - It describes "one" or "many" objects associated with other object.
 - The multiplicity is specified at the end of the association link.

	Notation	Description
	1	Only one instance
	0...1	Zero or one instance
	*	Many instances
	0...*	Zero or many instances
	1...*	One or many instances
	2...10	You can specify the int range

- The difference between two terms - the multiplicity & cardinality.
- The multiplicity is the constant constraint on the collection of the associated objects whereas the cardinality is the count of objects in that are in collection.

e.g.

(class diagram)

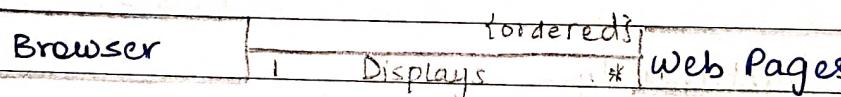


Q10. Define the following terms:

1) Ordering

This type of association is used to denote the set of objects at one end must appear in some specific order.

e.g.



Web browser displays the web pages in some specific order (First come first serve).

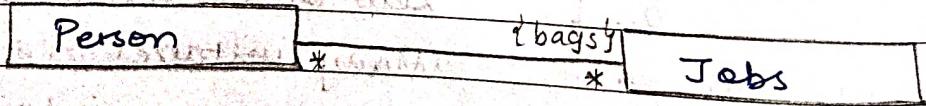
Hence the 'ordered' keyword is used at association end.

2)

Bags

- bag is a collection of objects at the other end that are duplicated or appear in some sequence.
- If the word bag appears at the other end, then the object at that end can appear more than once.

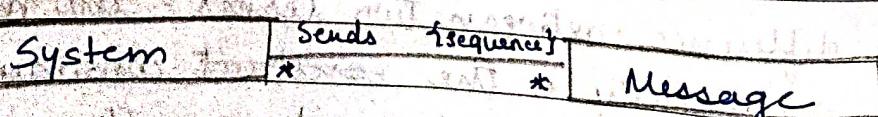
e.g.



3) Sequence

- Sequence is a collection of objects at the other end that are duplicated or appear in some sequence.

e.g.



4) Association class.

The abstract class is a class that allows an association to be a class itself.

- When two classes are related to each other by an association link, then association itself can have attributes & operations.
- Hence this association can be represented by a class.
- Eg - In the given illustration, Enrollment is association class for classes Student & Course.

Student	Enrollment		Course
- Name	- Stud ID	+ Enroll For	- CourseName
- Address	*	*	- CourseID
+ checkEligibility()			- Fee
+ SelectCourse()			+ AllocateStudent()
			+ CancelAllocation()
			+ UpdateCourse()

	Enrollment	Student
- Marks	+ calculate Avg. Marks()	

5) Qualified Association

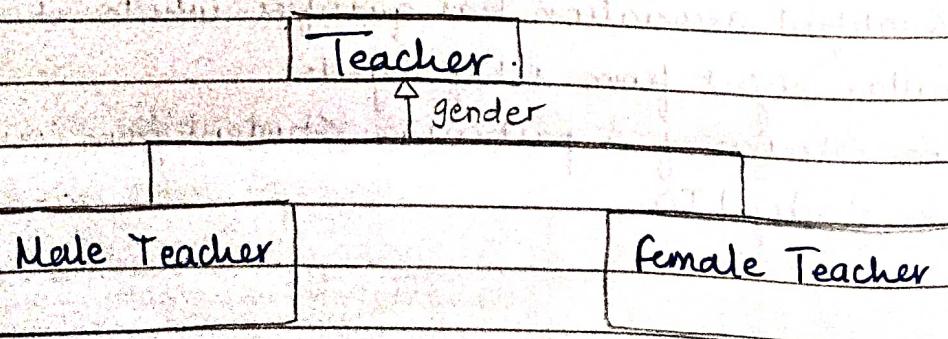
- The Qualified association has qualifier which is used to select particular object from set of objects.
- Eg - For selection of a particular Student from course, Qualifier will be student ID.

Course	Student
StudentId: Int	
1	1

- A qualifier is a property which defines selection key.

Generalization

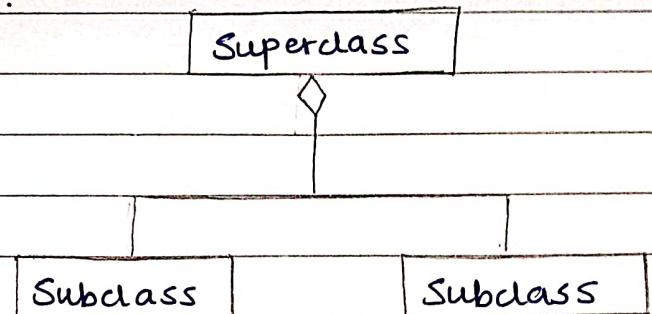
- Generalization is the relationship between parent a superclass & a subclass.
- A subclass inherits the features of superclass.
- The superclass is more general & it holds common attributes such as association, attributes & operations.
- The subclass is more specific.
- The subclass inherits general properties, attributes from superclass but also has its own specific attributes, operations & associations.
- Generalization is also called is-a relationship.
- This is because, each instance of subclass is an instance of the superclass.
- Generalization can be represented using arrow having hollow arrowhead.
- The arrow head points towards the superclass.
- The classes from which subordinate classes inherit the properties are called ancestors.
- The classes that use the properties of superclass are known as descendants.
- Generalization increases the flexibility of software.
- We can add new subclass & that subclass will automatically inherit attributes of the superclass.
- Eg. In the following figure, gender is generalisation set name?



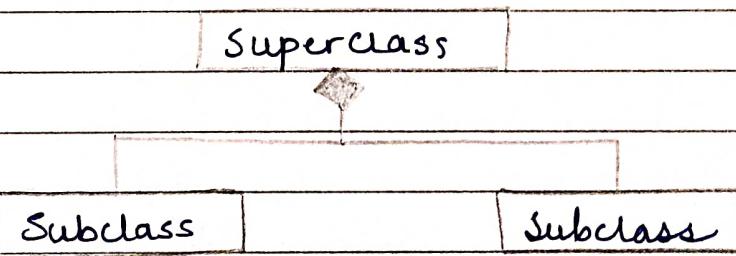
- The depth of the generalization hierarchy must not be too deep.

* Aggregation

- Aggregation is an extra-ordinary form of association relationship.
- Fundamentally, aggregation indicates ownership along with relationship amongst lifelines.
- Aggregation is also called "owns a" relationship or "part/whole" relationship.



- Composition relation is same as aggregation relationship.
- Only difference is aggregation is shown by a hollow diamond whereas composition is shown by a filled diamond.
- Composition is type "part/whole" relationship



* Dependency

- A dependency relationship shows relationship / connection amongst two or more elements.
- When there is change in one element it affects the other element.
- Using dependency relation is change in description of a particular element may affect the other element.
- Represented as

