



practical NO :- 04

Aim :- To draw the structural view diagram :
class diagram , object diagram

Theory :-

class diagram :-

- It is a graphical representation for describing a system in context of its static construction.
- class diagram contains the system classes with its data members, operations and relationship between classes.

Elements in class diagram
class

A set of objects containing similar data members and member functions is described by a class. In UML syntax, class is identified by solid outline rectangle with three compartments which contain

- class name :- A class is uniquely identified in a system by its name. A textual string is taken as class name. It lies in the first compartment in class rectangle.

- Attributes :- property shared by all instances



of a class. It lies in the second compartment in class rectangle.

- Operations :- An execution of an action can be performed for any object of a class. It lies in the last compartment in class rectangle.

Example :- To build a structural model for an Educational organization, 'course' can be treated as a class which contains attributes 'courseName' & 'courseID' with the operations 'addcourse()' & 'removecourse()' allowed to be performed for any object to that class,

Course
- courseName : string
- courseID : string
• addcourse()
• removecourse()

- Generalization / Specialization :- It describes how one class is derived from another class. Derived class inherits the properties of its parent class.

Example :- Geometric shapes is the class that describes how many sides a particular



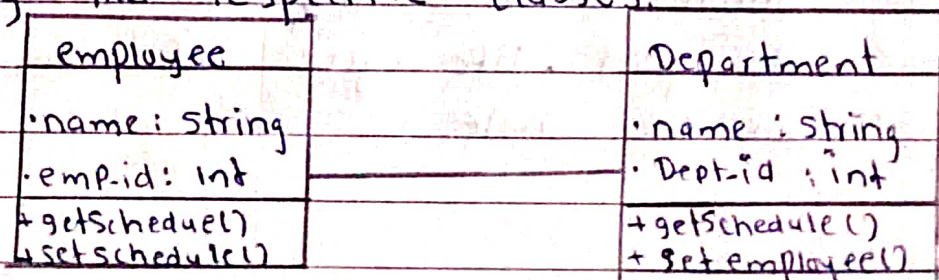
Shape has, Triangle, Quadrilateral and pentagon are the classes that inherit the property of the Geometric shapes class. So the relations among these classes are generalization.

Relationships

Existing relationship In a system describe legitimate connections between the classes in that system.

- Association :- It is an instance level relationship that allows exchanging messages among the objects of both ends of association. A simple straight line connecting two class boxes represent an association. we can give a name to association and also at the both end we may indicate role names and multiplicity of the adjacent classes. Association may be uni-directional.

Example :- In structure model for a system of an organization an employee is always assigned to a particular department and the association can be shown by a line connecting the respective classes.

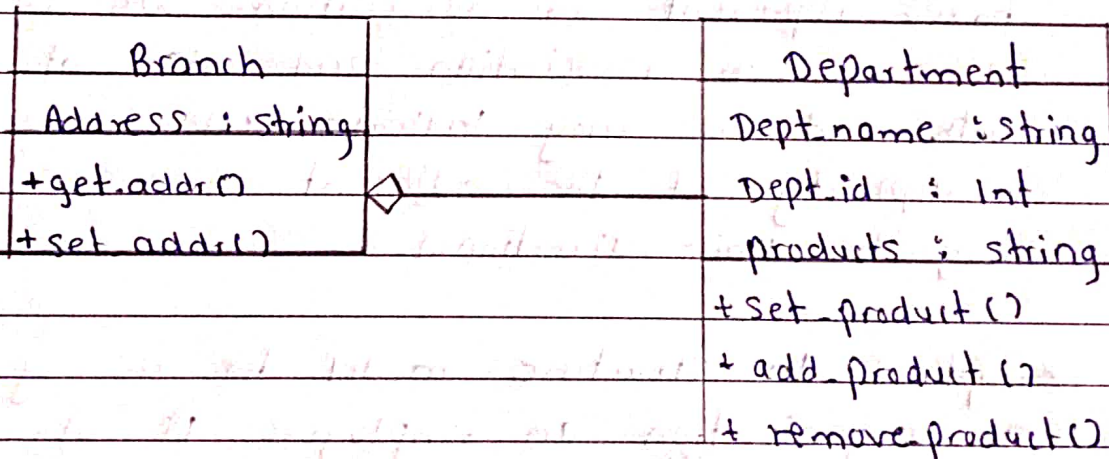




- **Aggregation**

- It is a special form of association which describe a part-whole relationship between a pair of classes. It means, in a relationship, when a class holds some instances of related class, then that relationship can be designed as an aggregation.

Example :- For a Supermarket in a city, each branch runs some of the departments they have. So, the relation among the classes 'Branch' and 'Department' can be designed as an aggregation.



- **Composition** :- It is a strong form of aggregation which describe that whole is completely owns its part. Life cycle of the part depends on the whole.



Example :

Shopping mall	Branch
Name : String	Address : String
+ getName()	+ getAddress()
+ setName()	+ setAddress()

multiplicity : It describes how many numbers of instances of one class is related to the number of instance of another class in an association.

Notation for different types of multiplicity :-

- Single Instance - 1
- Zero or one Instance - 0..1
- Zero or more Instance - 0..*
- One or more Instance - 1..*
- particular range - 2..6

Object diagram

- object diagrams are derived from class diagrams. So object diagrams are dependent upon class diagrams.
- object diagrams represent an instance of a class diagram.



- The basic concepts are similar for class diagram and object diagrams.
- object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moments.
- object diagram are used to render a set of object and their relationship as an instance.