

Unified Modeling Language Design (UML)

* OOA

Object oriented Analysis:-

Object oriented Analysis is a process that groups items that interact with one another, typically by class, data or behaviour, to create a model that accurately represents the intended purpose of the system as a whole.

Use case diagram

Class diagram

Sequence diagram

Collaboration diagram

State diagram.

Goals (Task)

- 1) Identifying objects (object relationship)
- 2) Organizing the objects by creating object model diagram
- 3) Defining the internals of objects or object attributes
- 4) Defining the actions of objects in object actions
- 5) Describing how the objects interact with the system

9845602870

OOD → complete description
Object Oriented Design.

1. In OOD, concepts in the OO analysis model, which are technology-independent, are mapped onto implementing classes, constraints are identified and interfaces are resulting in a model.

A detailed description of how the system is to be built on concrete technologies.

Implementation includes

1. Restructuring the class data.
2. Implementation of methods i.e., internal data structures and algorithms.
3. Implementation of control
4. Implementation of associations

OOP

Object oriented programming is a programming paradigm based upon objects (having data and methods) that aims to incorporate the advantages of modularity and reusability. Objects, which are usually instances of classes are used to interact with one another to design applications and computer programs.

OOA + OOD + OOP together called as object oriented modeling & design (OOMD)

Features of OOP.

- Bottom-up approach in program design.
- Programs organised around objects, grouped in classes.
- Focus on data with methods to operate upon objects data.
- Interaction b/w objects through functions.
- Reusability of designs through creation of new classes by adding features to existing classes.

Object

A real word element in an OO environment that may have physical or conceptual existence.

- Identity that distinguishes it from other objects
- State determines the characteristic properties of an object as well as the values of the properties that the object holds.
- Behaviour that represents external actions.

Classes

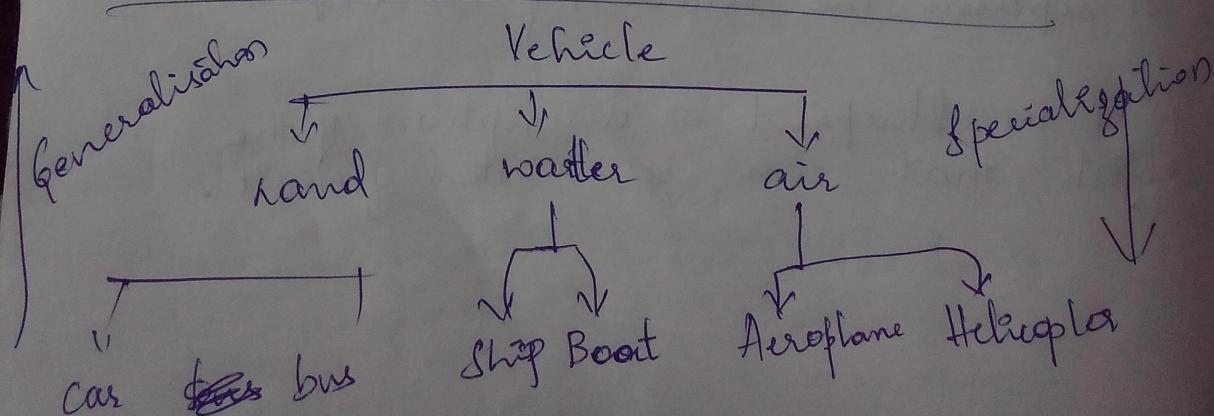
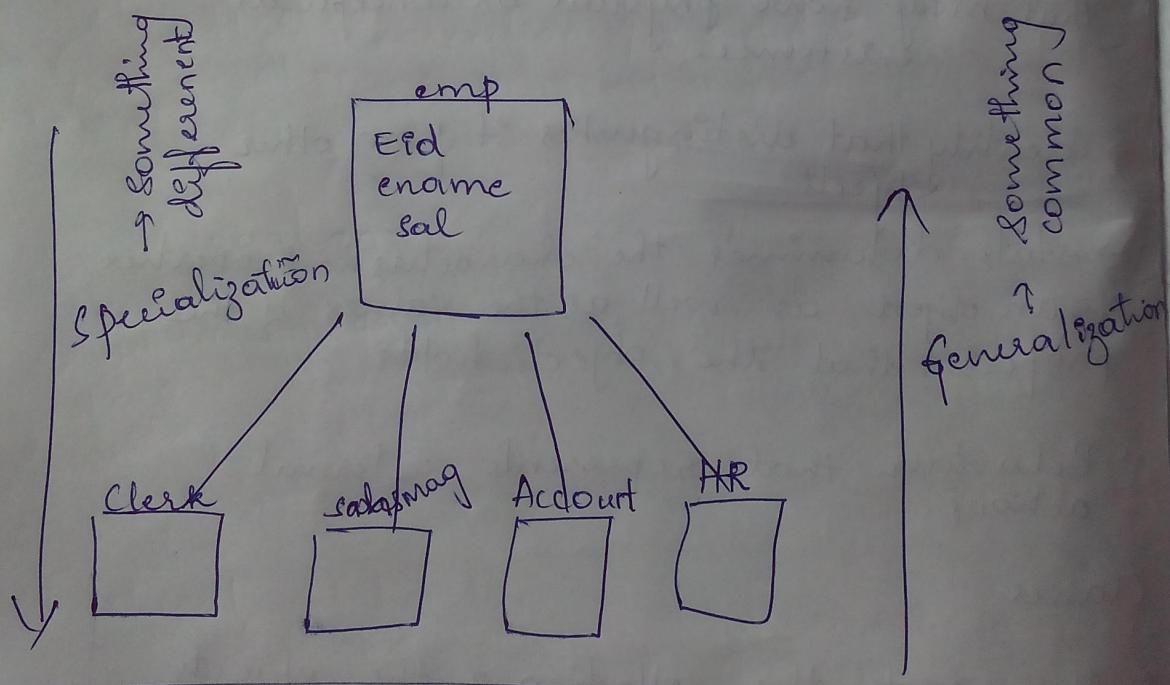
- It represents the collection of the objects
- Gives the blueprint or description of the objects that can be created from it.

Instantiation

Creation of an object as a member of a class is called instantiation. Thus object is an instance of class.

Generalization

In the generalization process, the common characteristics of class are combined to form a class in a higher level of hierarchy, i.e., subclasses are combined to form a generalized super-class. It represents an "is-a-kind-of" relationship. For example, "car is a kind of land vehicle" or ship is a kind of water vehicle.



Specialization

It is the reverse process of generalization.
Here, the distinguishing features of groups of objects are used to form specialized classes from existing classes.
It can be said that the subclasses are the specialized versions of the super-class.

UML

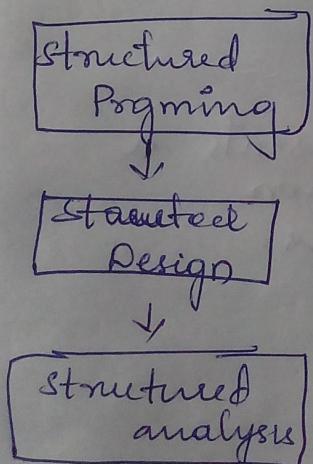
Modeling

- Describing a system at a high level of abstraction.
 - A model of the system.
 - Used for requirements and specifications

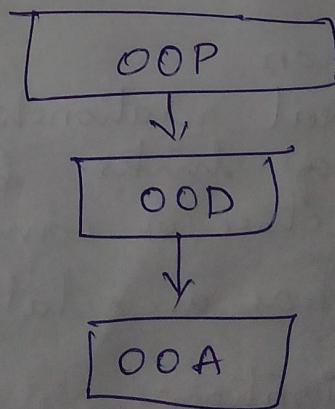
OODM

Evolution of OO Development methods.

70/80ies



80/90ies



What is UML?

$$15 \times 2 = 30 \times 30 \\ = 900$$

- unified modeling language.
- It is a industry-standard graphical language for specifying, visualizing, constructing and documentation the artifacts of software systems.
- The UML uses mostly graphical notations to express the OO analysis and design of software projects.
- Simplifies the complex process of software design.

Relationship

- Dependency → Modification of one object affects the other object.
- Association → connection b/w 2 objects.
- Generalization → is a relationship
- Realization → signing authority.

Association

Composition Aggregation
(strong association) (weak association)

Association

Structural relationship that represents a group of links having common structure or common behavior.
i.e; has a relationship.

Generalization:-

relationship where sub-class inherit structure & behaviour from super classes.

Realization

This is a semantic relationship b/w two or more classifiers such that one classifier lays down a contract that the other classifies enure to abide by.

Diagrams

A diagram is a graphical representation of a system. It comprises of a group of elements generally in the form of a graph.

Imp. Types of UML diagrams

use case diagram

class diagram

sequence diagram

collaboration diagram

state diagram

These are some imp type of UML diagram.

UML Notations

- Class
- Object
- Interface
- Package
- Relationship

1) Class. represented in
Rectangular box. having

3 sections

→ Name of class. (top section)

→ Attribute of the class. (Middle section)

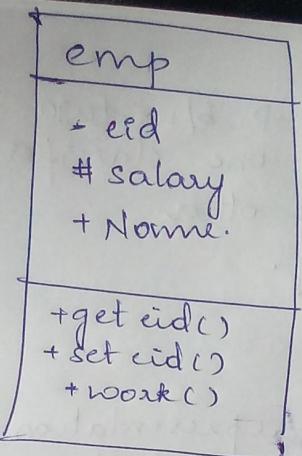
→ Behavior of the class. (Bottom section)

For attributes.

+ for public # protected.

- for private

Ex



→ top section

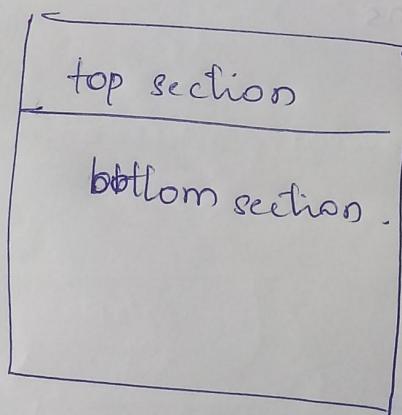
→ Middle section

→ Bottom section.

2) Object

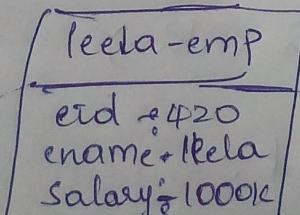
The top section contains the name of the object with the name of the class or the package of which it is an instance of. The name takes the following forms.

- Object name - class name
- Object name - class name :: package name
- Object name - in a case of anonymous objects.



The bottom selection represents the values of the attributes. It takes the form attribute - name = value.

Sometimes objects are represented using rounded rectangles.



Depend
Associate
Direct
Associate
Inheritanc
Realizat
Aggrego
Composit

Relationship .

