Sameep Dhakal
ACE074BCT063

| OO Based Design Cycle | Conventional Design cycle |
|---|---|
| 1) System is divided into objects making software more modular since different objects of class can be created in order to fulfill multiple tasks with same class. Example: using buy class creating objects everything new user wants to buy stuff helps to the same task seperatey without having values changed. | System is divided into functions making software less modular since for every task to be completed a program has to call functions which interm may change whole value. example: If someone wants to buy but then the price has to be set and for each transaction program has to wait for that transaction to complete so that it can overwrite that price variable. |
| 2. functions and data are modelled in the class and that data can be accessed by the objects of that class only hence making it more efficient and secure. | Function and data are modelled seperately since that the data has to be accessed by multiple functions in order to run whole program which is not efficient and variable data is not secure |
| 3) System components are independent of each other. and Variables can be modified | System components are dependent of each other. and variables can't be modified |
| 4) Inheritance and Polymorphism are possible. | Inheritance and polymorphism are impossible. |

Sameep Dhakal
ACE074BCT063

| | |
|---|---|
| 5. No restriction for reuse with the help of inheritance and objects one ~~can~~ be code can be raused rather easily | ~~Results~~ Reuse is limited since the function can perform particular task and if to do similar task with different parameters |
| 6. It reflects real world entity. example. if we have to represent a guitar then guitar class with functions and variables which help to represent real world objects like guitar more effectively And another entity a similar task can be done | It doesn't reflect real world entity. example'. To emulate a guitar it's complicated since functions only are not enough to represent it and variables can be accessed by other functions too changing its originality hence making it harder to represent |
| 7. Development process is iterative and incremental since features like classes can be added one at a time testing / debugging and then increased to add more classes since they are independent of each other hence adding more features into the program. | Development process is linear since functions have to be added in chunk for the program to work and further adding functions into it are complicated and require through planning and if bug occurs it becomes complicated to do unit tests and check error. |

Sameep Dhakal

ACE074BCT063

Q.NO.2 =>.

Functional Requirements.

-> Functional requirements define a function of a system or its component, where a function is described as a specification of behaviour between outputs and inputs. As defined in requirements engineering, functional requirements specify particular result of program

a) It describes the interaction between the system and its environment

b) Typically, functional requirements will specify a behaviour or function, for example " Display the name, total size, available space, and format of a flash drive connected to USB port

c) Some of more typical functional requirement include

- Business Rules.
- Authentication
- Historical Data
- Administrative functions
- Audit Traking
- External interfaces

Examples:-

System shall communicate with external system X, what conditions must be met for a message to be sent

Non functional Requirements.

It describes how the system works or behave and that it is a constraint upon the systems behaviour

-> It cover all remaining requirements which are not covered by functional requirements.

Sameep Dhakal
ACE074BCT063

They specify criteria that judge the operation of the system, rather than specific behaviours.

example:
Modified data in database should be updated for all users using it within 2 seconds.

Some typical non-functional requirements are
- Perfomance
  Scalibility
  Availability
  Maintainability
  Security
  reliability.

→ ~~Funct~~ Non functional requirements ~~require~~ describe a restriction or constraints that units our choices.

Example:
Paychecks distributed no more than four hours after initial data is read

Q.NO.3 ⇒.

Agile methodology is the type of project management process. It anticipates change and allows fore flexibility. The major strengths are described below.

1) Quality Assurance.
It is assured by the testing team and since development is conducted in short cycles, testing can be done non-stop allowing us to produce a good final product

ii) constant interaction with stakeholdes
constant interaction with each member of the team and with customers helps us to avoid producing tons of technical documents, process and tools.

ii) Flexible
Short cycles and constant iteration allows us to adapt our project frequenctly and tailor it to the customers need at any moment. Working projects can be delivered quicker hence changing features with interaction make it easier

iv) customer satisfication Apriority.
Since this method contains close coordination with customer, hence the customer has larger impact on development process. Customer feedback is always taken to make changes.

Requirements elicitation Process in OOAD are
a) Interviews: Interviews are strong medium to collect requirements

b). Questionaire: A document with predifined set of objective questions and respective question is handed over all stakeholders to answer.

c) Survey:
organizations may conduct surveys among various stakeholders, ~~to~~ ~~answer~~, ~~which~~ ~~are~~ querying about their expectations and requirement in upcoming system.

Sameep Dhakal

d) Domain Analysis.

Every software fall into some domain category. The expert in each domain can provide great help to analyze general and specific requirements

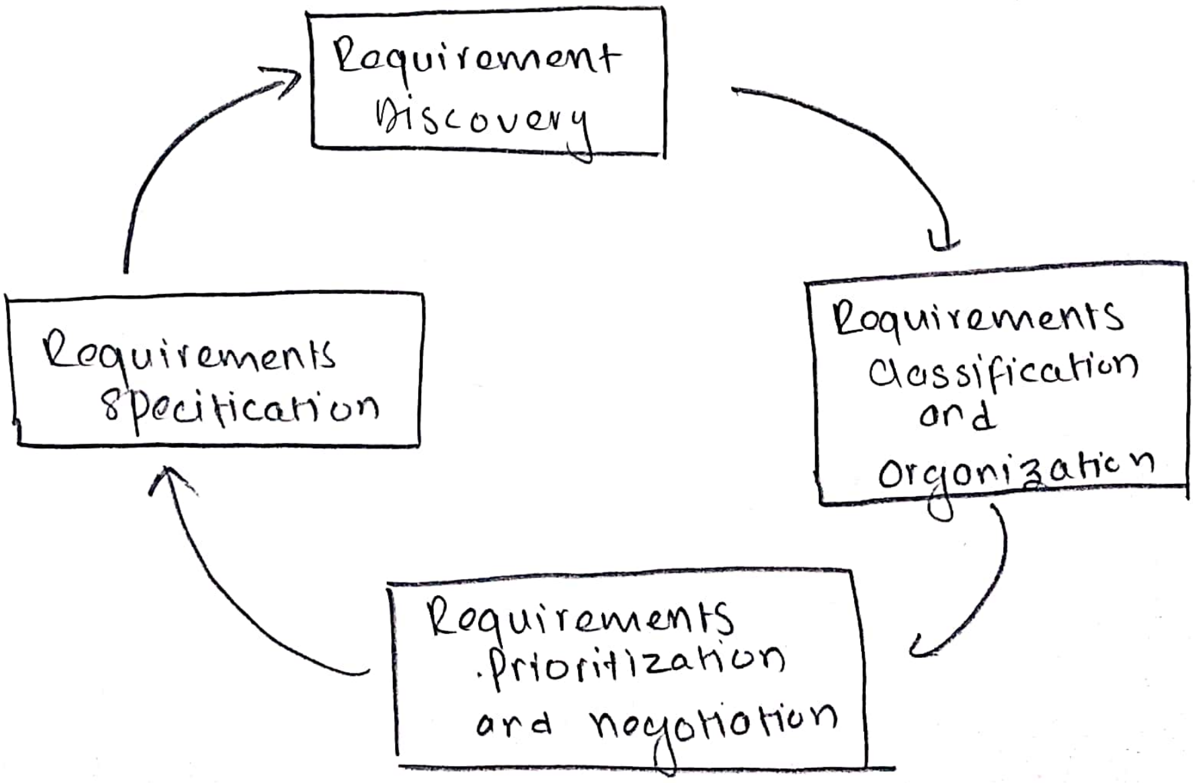
e) Observation.

The actual working of existing similar type of installed components can be observed.

f) Brainstorming

A spontaneous group debate to produce ideas and ways of solving problems

These all things can be described in a diagram in main 4 processes.

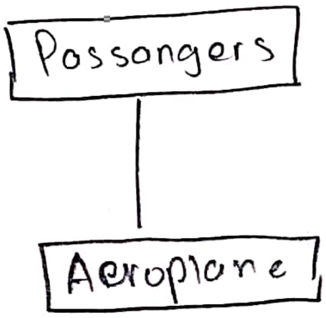


Requirement Discovery
Requirements Classification and Organization
Requirements Prioritization and negotiation
Requirements Specification

Q.No 4 Conceptual class is an idea thing or object that illustrates the requirements in the software we are building
→ It provides all the necessities of problem
It is considered in terms of symbols, intension and extention.

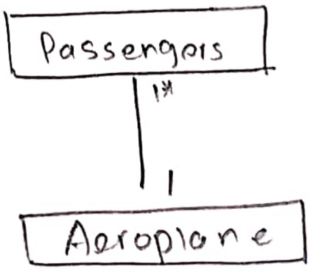
Domain model is generally implemented as an object model within a layer that uses a lower level layer for persistences and publishes an API to a higher level layer to gain access to data and behaviour of the model. In UML, a class diagram is used to represent domain model.
→ It contain conceptual class, association between conceptual classes and attributes of conceptual class.

The various Relationships implemented in class diagram are.
1) Assotiation
Assotiation encompasses just about any logical connection or relationship between classes.



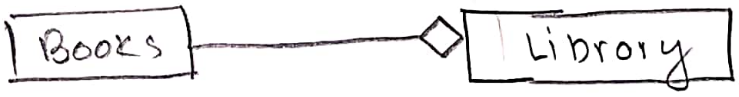
Sameep Dhakal

Multiplicity

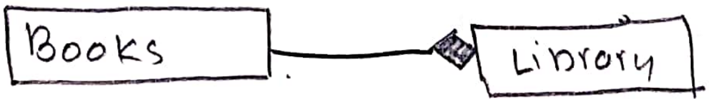


It is active logical assotiation when cordinility of Class is requrred.

Aggregation
It represents the part of relationship.

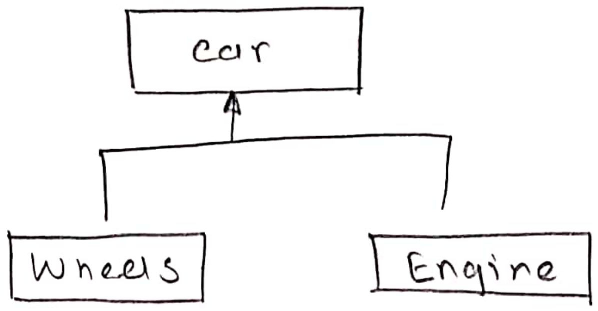


Composition.
Type of aggregation when parts are destrayed. when the whole is destroyed.
A solid line filled diamond shape is used for representation.



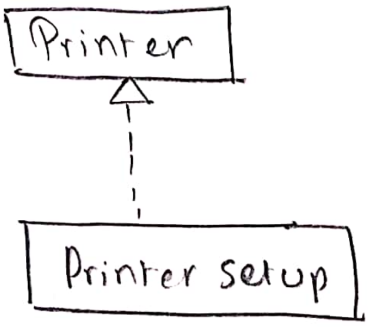
Generalization

It represents "is-a" relationship. Subclass 1 and 2 are specialization of Superclass. Solid line with arrowhead at that point

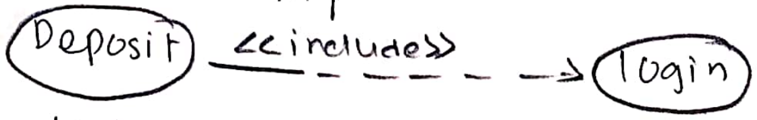


Dependancy.

Exist between two Classes if the changes to definition of one may cause changes to other. dashed line with open arrow.



include relationship

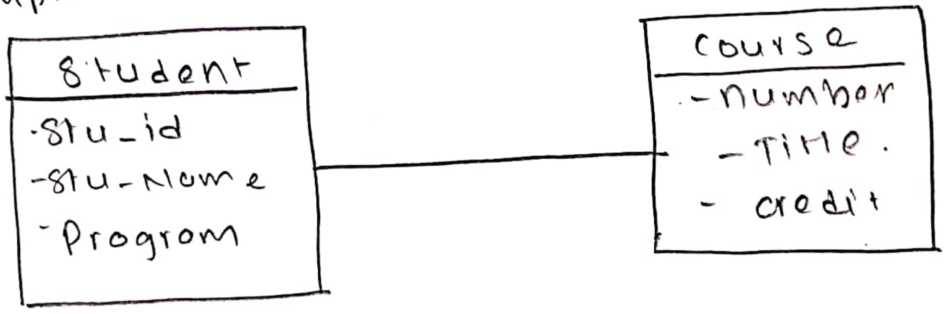


if login is must for deposit we use this

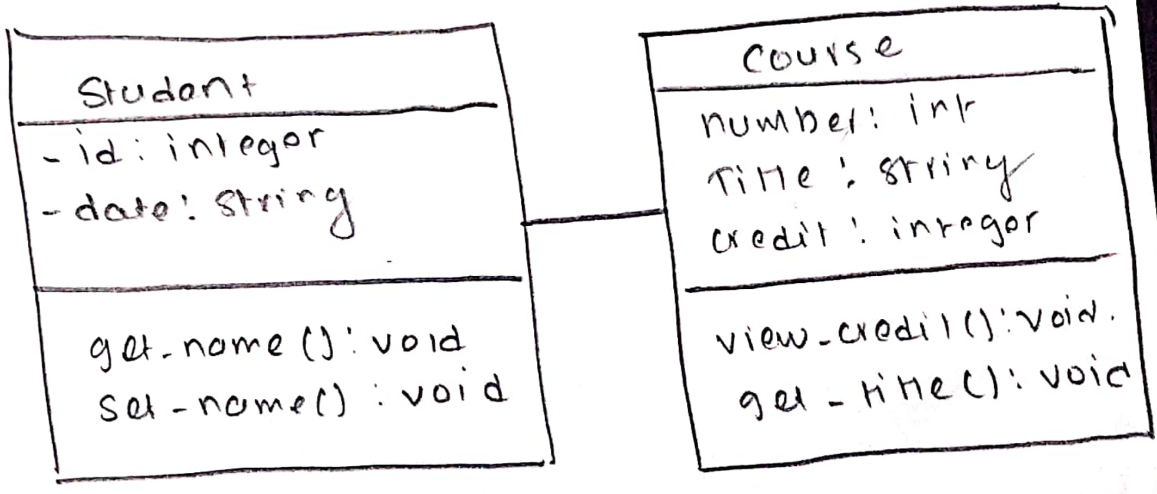
Q. No. 5 =),
The vorious Models in OOAD are:

Conceptual model
It is organized and structured knowledge of ~~progra~~ problem. It illustrates only the noteworthy concepts in domain.
It shows what should bo done in a program.

example.

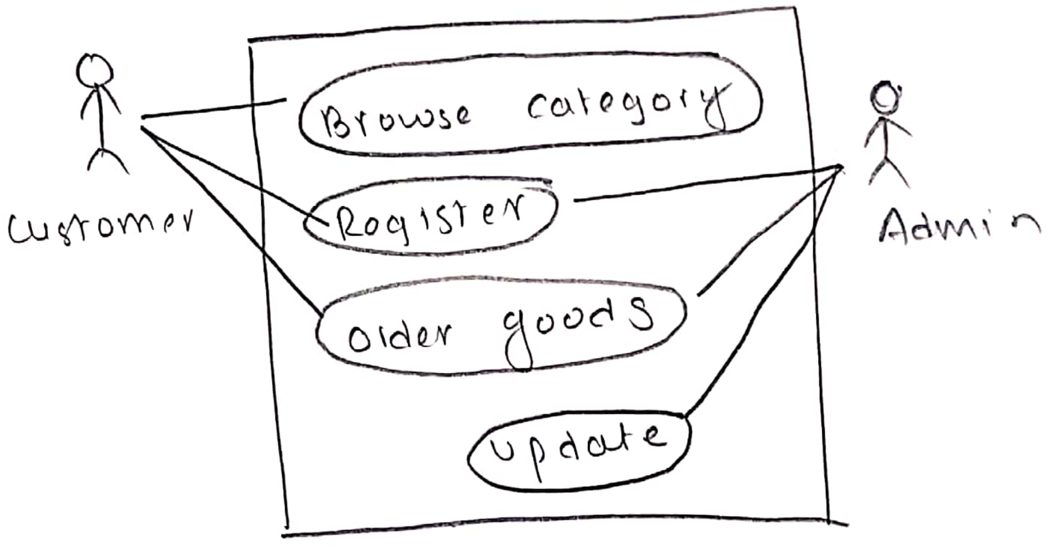


Structural model.
It helps to capture static features of the system. It never describes dynamic features. It is framework where all component exist, it is sent to coding.



Sameep Dhakal

Behavioral diagram.
Such as activity, usecase diagram It is used to describe interaction of system. it represent interaction among structural diagrams. It show dynamic nature



Use case of online shopping.