

Units: Object-oriented Development

DATE

* Introduction

- In object oriented approach, the focus is on capturing the structure and behavior of information systems into small modules that combines both data & process.
- The main objective of Object-Oriented design (OOD) is to improve the quality & productivity of system analysis and design by making it more usable.

→ The difference between Object oriented development and structured development is that:

Structured development is a method of software development that emphasizes breaking down the program into smaller, modular components that can be easily understood & managed. It follows a top-down approach & is based on structured programming principles such as top-down design, stepwise refinement & structured coding.

On the other hand, Object-oriented development (OOD) is a method of software development that emphasizes creating objects that contain both data & behavior. It follows an object-oriented programming (OOP) paradigm & is based on OOP principles such as encapsulation, inheritance, & polymorphism. OOD also focuses on the objects & their interactions within a program rather than just the procedures.

In summary, structured development focuses on procedural breakdown of a program, OOD focuses on the objects & their interactions within a program, making it more efficient & flexible to design & develop complex systems.

Elements of object-oriented system

- I) Object: An object is a real-world element in an object-oriented environment that may have a physical or a conceptual existence. All tangible entities (student, patient) and some intangible entities (bank account) are modeled as objects.
- II) Attributes: They describe the information about the object.
- III) Behavior: It specifies what the object can do. Defines the operations performed on objects.
- IV) Class: A class encapsulates the data & its behavior. Objects with similar meaning & purpose are grouped together as classes.
- V) Methods: They determine the behavior of a class.
- VI) Message: A message is a function or procedure call from one object to another.

Features/Characteristics of object-oriented system

- I) Encapsulation:
 - It is the process of information hiding.
 - It is simply the combination of process & data into a single entity.
 - Data of an object is hidden from rest of the system & available only through the services of the class.

P.P) Abstraction

- Classes are built on the basis of abstraction, where a set of similar objects are observed & their common characteristics are listed.
- Data abstraction refers to providing only needed information to the outside world and hiding implementation details.

P.P) Inheritance

- It allows to create sub-classes from an existing class by inheriting the attributes and/or operations of the existing class.

P.V) Polymorphism

- It is the ability to make more than one form. Any operation may show different behaviors in different scenarios.

V) Reusability

- The classes once defined can easily be used by other applications. Once a class is defined it can be used again & again by defining object of the class type.

* Object-Oriented system analysis & design (OOSAD)

It is the method of analyzing and designing a system using object-oriented concepts like class, objects, & so on. This approach emphasizes understanding the problem domain in terms of objects & their interactions, rather than just focusing on processes & procedures. This methodology encourages re-use. New applications can use existing ~~modules~~, thereby reducing development cost & cycle-time.

X. Unified Modeling Language (UML)

- The UML is a general purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.
- It is designed to enable users to develop an expressive, ready to use visual modeling language.
- In addition, it supports high level development concepts such as frameworks, patterns & collaboration.
- UML includes a collection of elements such as:
 - Programming Language statements
 - Actors: Specify a role played by a user
 - Activities: tasks which must take place in order to fulfil an operation contract.
 - Business process: includes a collection of tasks
 - Logical & Reusable Software Components
- UML can be divided into two categories:
 - ① Structure diagrams → class diagram, object diagram
Component diagram & deployment diagram
 - ② Behavioral diagram → use case diagram,
activity diagram, sequence diagram & state diagram

* Structural diagram

They are used to represent the static structure of a system. They provide a visual representation of the class, objects, interfaces & relationships. Mosty used structural diagrams are:

1. Class diagram

- It represents the classes & their relationships, attributes and methods.
- It consists of class, attributes and list of operations that can be performed on the class.
- The purpose of class diagram is to represent the static view of the system.
- Class diagrams are the only diagrams which can be directly mapped with object oriented languages & thus widely used at the time of construction.

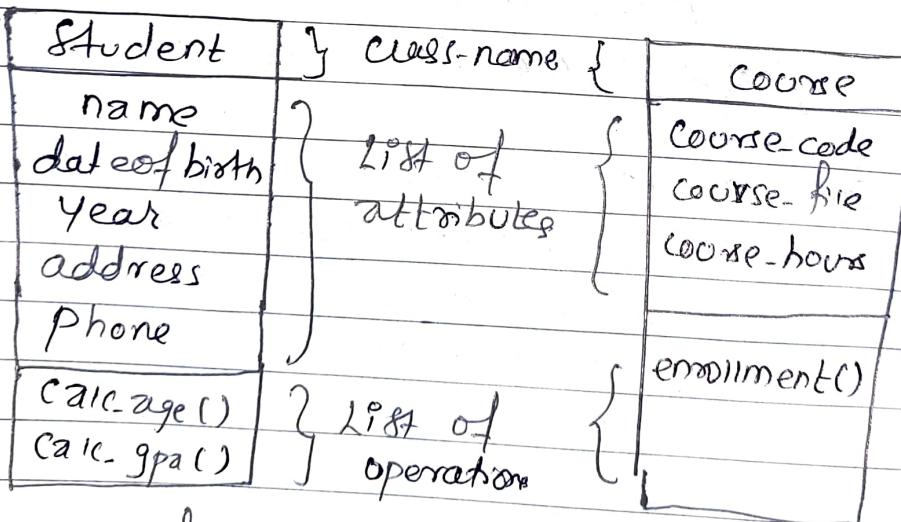
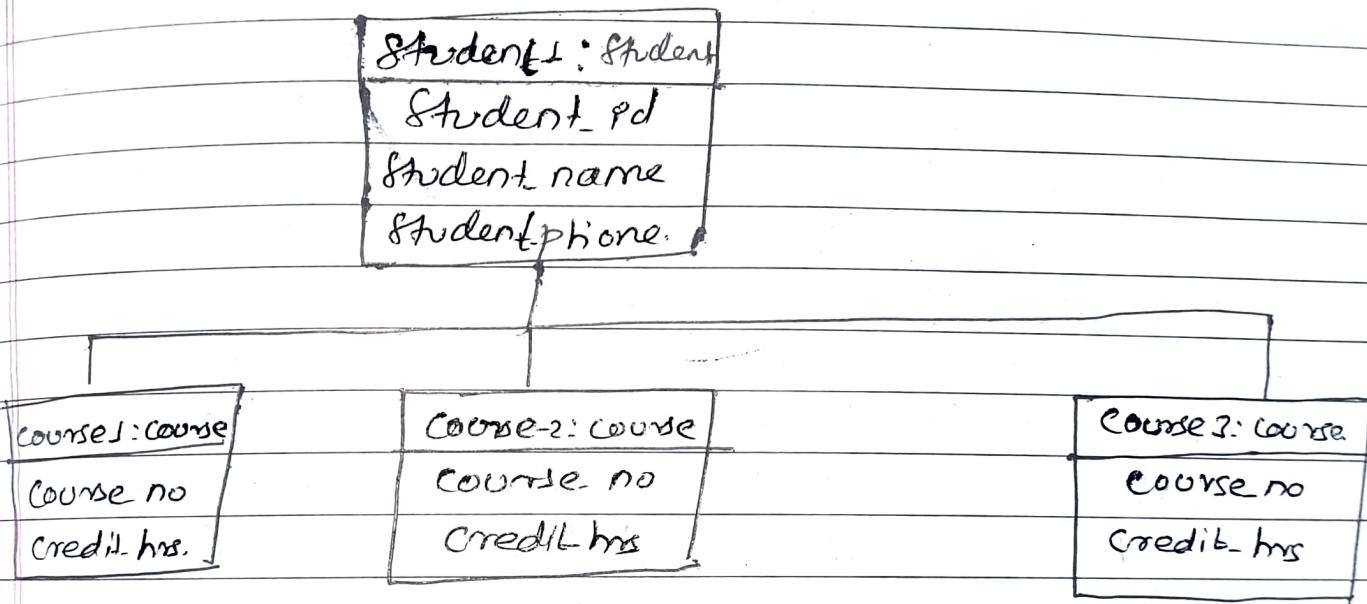


fig: UML class diagram showing two classes Student & course.

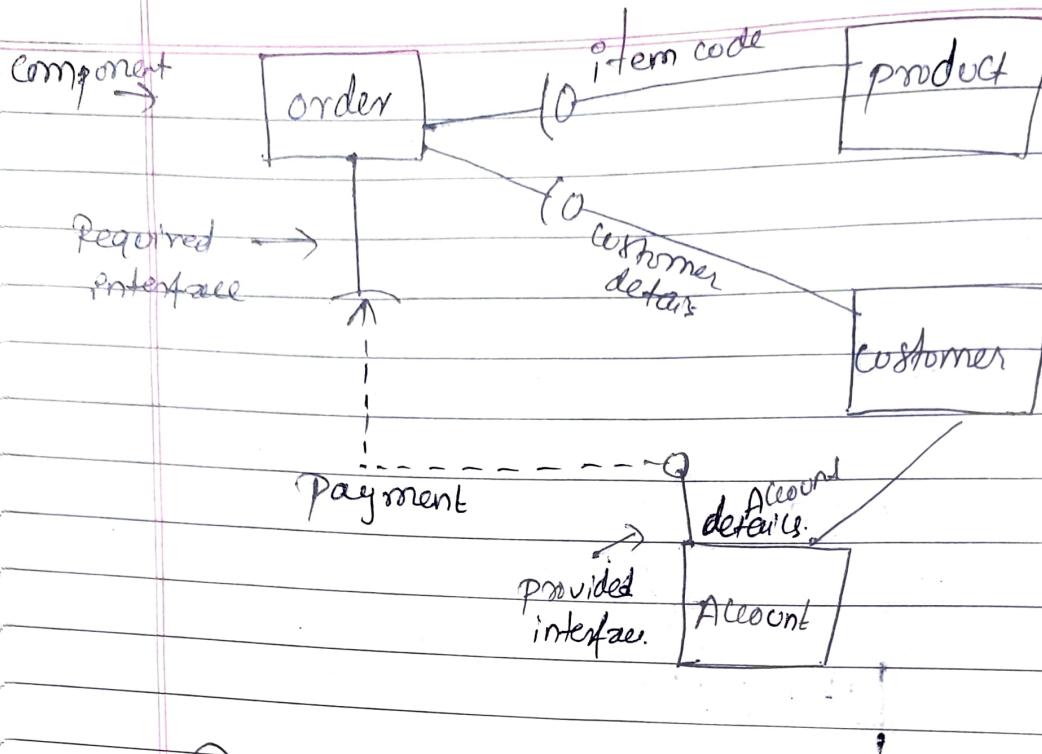
2. Object diagram

- Object diagrams consist of specific instances of classes & relationships between them at a point of time.
- An object diagram is similar to class diagram, except that it shows the instances of classes in the system.
- Object diagrams are derived from class diagrams so object diagrams are dependent on class diagrams.



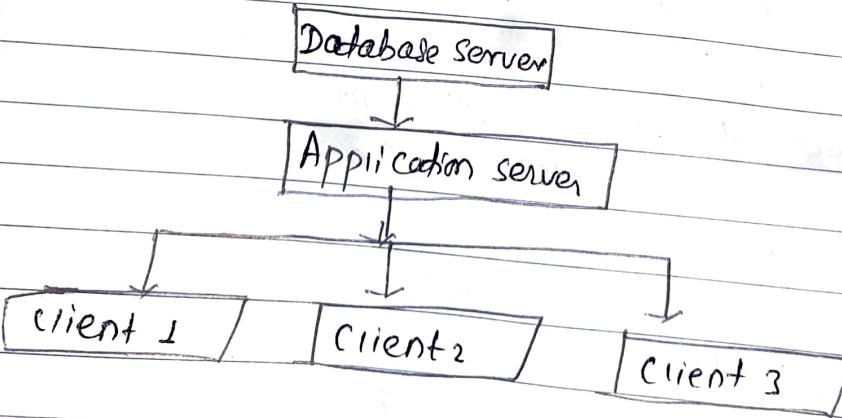
3. Component diagram

- It represents a set of components & their relationships.
- These components consist of classes, interfaces or collaborations.
- This diagram is used to represent how the physical components in the system have been organized.
- We use them for modeling implementation details.
- An example of component diagram is:



iv) Deployment diagram

- They are also known as the Implementation diagrams.
- These diagrams show the implementation environment of system and are used to represent system hardware & its software which are connected to each other.
- It has a set of nodes & their relationship. These nodes are the physical entities.
- It represents the physical architecture of the system



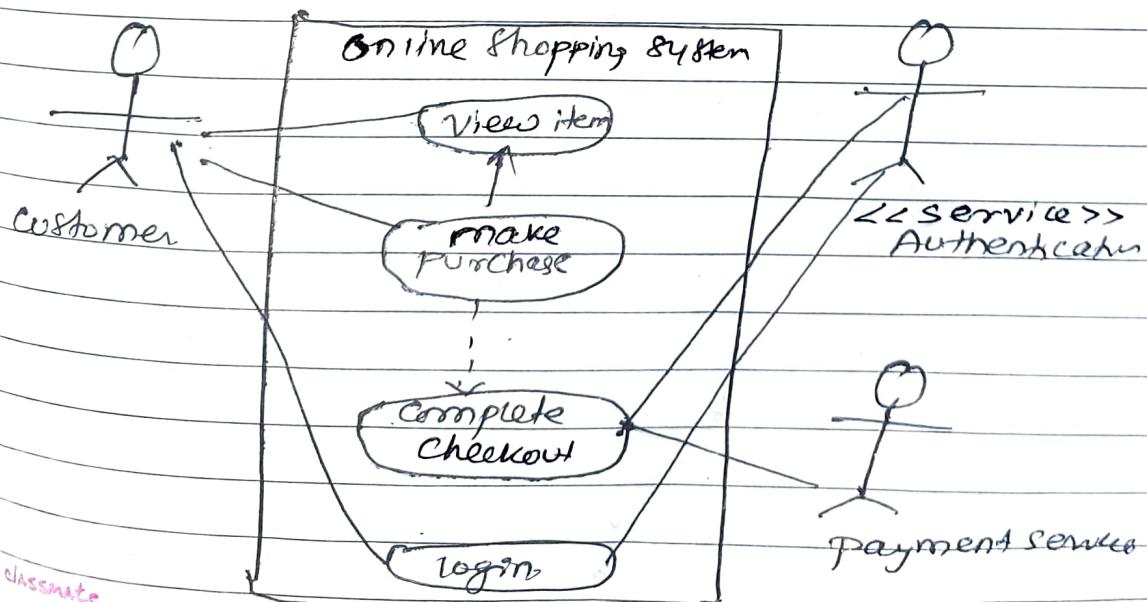
Behaviour Diagrams

A behaviour diagram in UML represents the dynamic behaviour of system, such as how objects interact with one another & how they change over time. It emphasises on the functionality of software system.

i) Use case diagram

- It represents a graphical overview of the actors involved in a system, different functions needed by those actors & how these different functions interact.
- Use case shows the interaction of things outside the system with the system itself & an actor represents the roles that the users of the use cases play.
- An actor may be a person (e.g. student, customer) or another system (e.g. bank, institution)

Ex: use case diagram of online shopping system



Q9) Activity diagram

- Activity diagram represents workflows in a graphic way, they can be used to describe the business workflow or the operational workflow of any component in the system.
- It is basically the flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
- Sometimes activity diagram are used as an alternative to state machine diagrams.

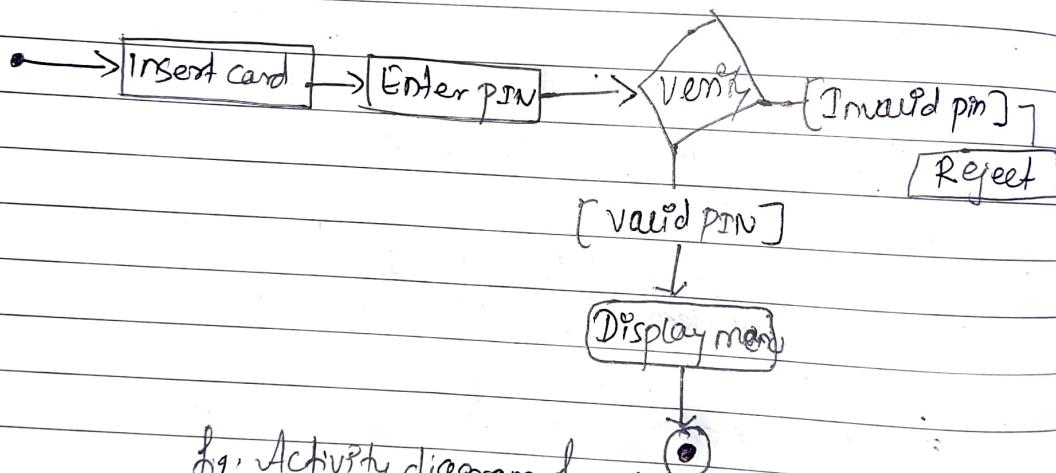


fig: Activity diagram for ATM machine

Q10) State diagram

- State diagrams are similar to activity diagrams, although notations and usage change a bit.
- They are sometimes known as State machine diagrams or State Chart diagrams as well.
- These diagrams are very useful to describe the behavior of objects that act differently according to the state they are in at the moment.

- The State diagram below shows the basic states & actions.

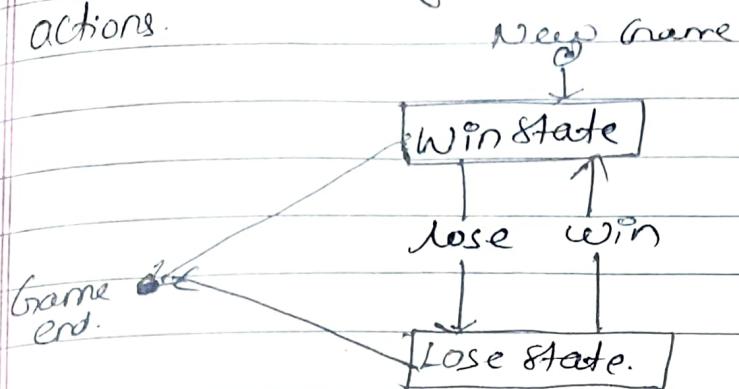


fig. State Diagram for a game

iv) Sequence diagram

- Sequence diagram shows how objects interact with each other and the order these interaction occur.
- They show the interactions for a particular scenario. The processes are represented vertically & interacting are shown as arrows.
- These diagrams are widely used by businessmen & software developers to document & understand requirements for new & existing system.

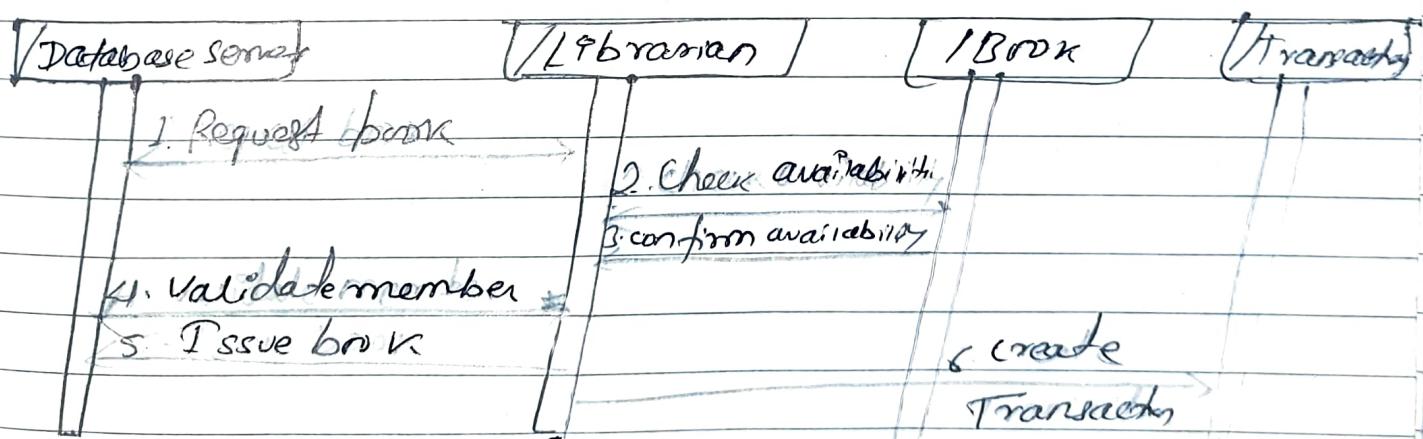


fig. Sequence diagram for Library management system

Questions asked

- Q. What is object oriented development? How is it different from structured development. (2016(Chew), -5 marks)
- Q. What is class diagram? Explain with suitable example. (2018 - 5 marks)