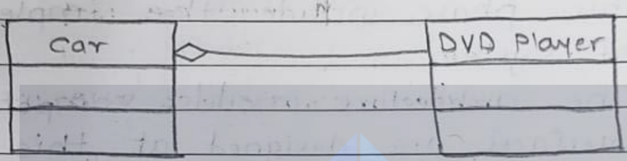
DATE: 9/8/2023
PAGE NO.:

Assignment 2

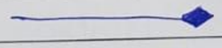
Q.1 What is aggregation and composition? Give their respective UML notation with an example.

⟶ 1] Aggregation is a type of association.
2] It is used to represent whole-part relationship
3] Aggregation passes has-a relationship
4] It is denoted as follows -



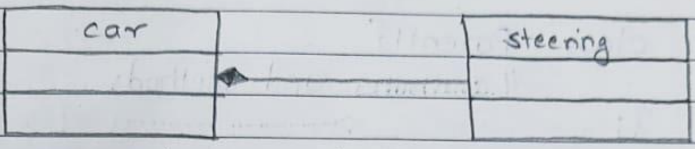
Example - Aggregation relationship

5] for example -
DVD player and Car are the two classes that can be associated by aggregate relationship. Note that DVD player can exists without car and car can be without a DVD player. We can read this relationship as "DVD player is a part of car".
6] Aggregation and composition are two forms of whole-part relationship. The composition is restricted aggregation. That is, when one object contains another object and if contained object can not exist without the existence of the container object, then it is called Composition
7] It is denoted as follows -



Composition

8] For example - steering and car are the two classes that are associated by the composite relationship. This is restricted relationship because a car can not exist without the steering.



Example - composite relationship

We can read the relationship as "steering is a part of car"

Q.2 What is multiple inheritance? Explain the kinds of multiple inheritance.

⟹ Multiple inheritance -

The Multiple inheritance is a kind of inheritance in which the derived classes are derived from more than one super classes. The class with more than one superclass is called join class.

* There are two main kinds of multiple Inheritance.

i) Pure Multiple Inheritance.

- In pure multiple inheritance a class inherits from multiple parent classes without any ambiguity mano
- The language compiler or runtime system manages the conflicts that might arise from multiple inherited methods or attributes.

languages like C++ support pure multiple inheritance.

Example-

```
class Parent1{
        // attributes and methods.
};
class Parent 2 {
      // attributes and methods
};
class child : public Parent1, public Parent2 {
        // attribute and methods
};
```

② Diamond (or Readly Diamond Inheritance):

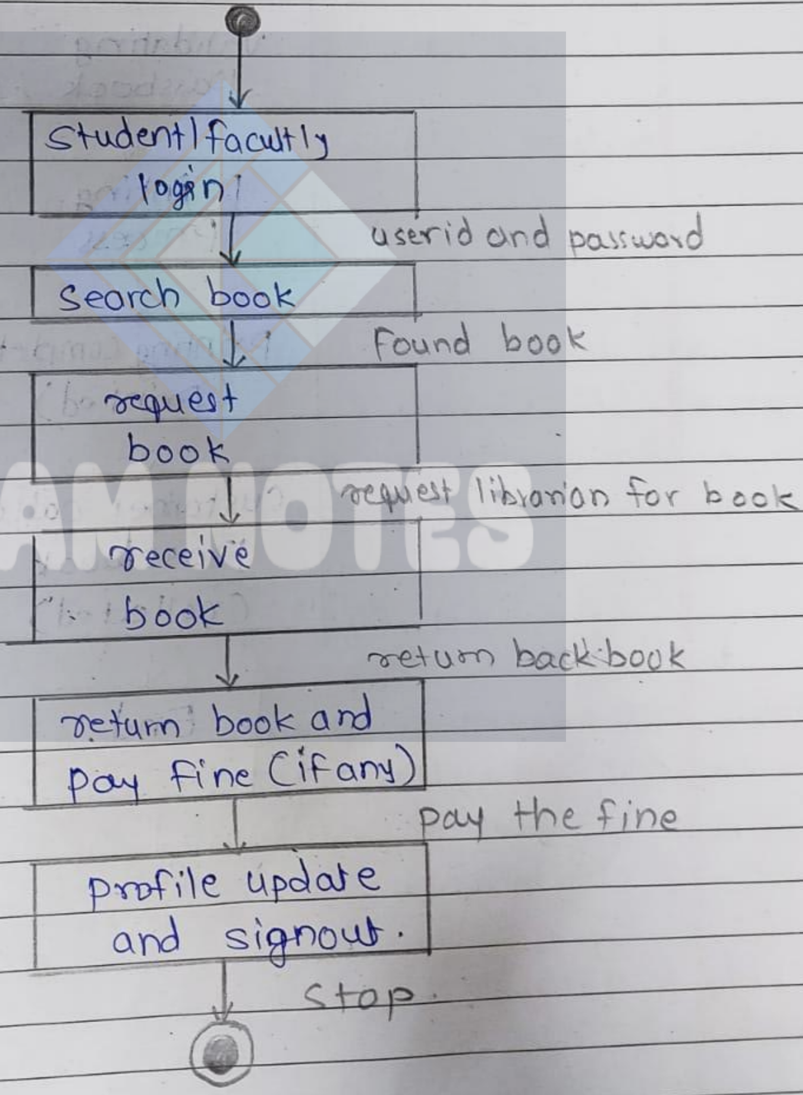
- Diamond inheritance occurs when a subclass inherits from two or more classes that have a common base class.
- Thes can lead to ambiguity and conflicts when the subclass tried to access member from the common base class.
- Language like C++ have mechanisms to resolve these conflicts, such as virtual inheritance.
- Example-

```
class grandparent{
      // attributes and methods
};
class Parent 1: public grandparent {
        // attributes and methods
};
```

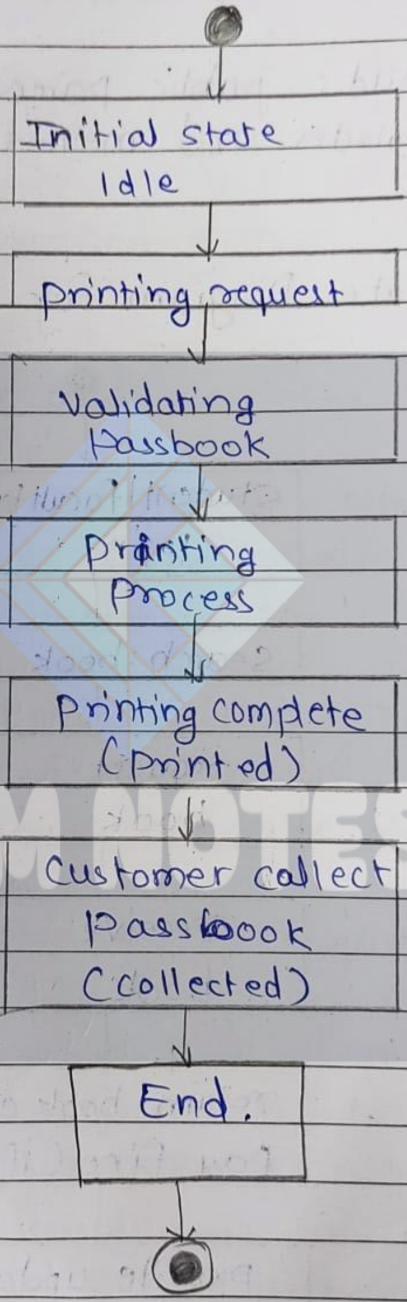
Cl
}
cl
};
Q.3. Dr

```
class Parent 2; public grandparent{
        // attributes and methods
 };
class child; public parent1, public parent 2{
     // attributes and methods.
};
```

Q.3. Draw state Diagram for library system.



Q.4. Draw state diagram for passbook printing in book.



(Q.5) What is state diagram explain the state in developing state diagram?

⟹ • An state diagram is a graphical representation used in software engineering and systems analysis to model the behavior of a system or an object over time.

• It depicts the various states that an object or system can be in and the transitions between those states in response to events or conditions.

• Developing a state diagram involves ~~souce~~ several steps:

① Identify States:-

• Identify the distinct states that the object or system can be in states should be defined based on the behavior and characteristics of the object or system.

② Define events :-

Determine the events or conditions that trigger transitions between states. Events should represent meanigful ~~no~~ actions or changes that affect the object's behavior.

③ Map Transitions:

Establish the transitions between states based on the identified events. Describe how the object or system responds to each event by transitioning to a new state.

④ Specify Actions :-

Define the Actions or behaviours associated with each transition. What should happen when the transition occurs? Specify any changes to attributes or interactions with other objects.

⑤ Add guard conditions:-

· If necessary, include guard conditions that restrict the occurance of transitions based on specific criteria. These conditions may include attribute values or other factors.

⑥ Draw the diagram:

· Use appropriate notation to draw the state diagram, including rounded rectangles for States, arrows for transitions, and labels for events and actions. Make to arrange the diagram in a clear and logical manner.

⑦ Review and Refine

· Review the state diagram to ensure it accuracy captures the desired behavior Of the object or system. Make any necessary adjustments or refinements based on feedback and analysis.