CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Sub:	Object Oriented Modeling and Design					Code:	10CS71
Date:	20/09/2017	Duration:	90 mins	Max Marks: 50	Sem: VII	Branch:	CSE

Note: : Answer any 5 questions   Total marks: 50

Question #		Description	Marks Distribution		Max Marks
1	a)	Definition of object orientation 5 stages in methodology	2M 6M	8M	10 M
	b)	Definition of Class model Example of Class Model	2M	2M	
2	a)	six points with explanation with diagrams Examples	1.5*6=9M 1M	10M	10 M
3	a)	Class diagram with 3 generalization, aggregation, multiplicity, association names	10M	10M	10M
4	a)	Definition of Model Explain any 4 Purpose each carries	2M 1*4=4M	6M	10 M
	b)	Definition of ordering ,bags, Sequence Examples	1*3=3M 1M	4M	
5	a)	Aggregation, Composition and Propagation of operation definition and Examples	2*3=6M	6M	10 M
	b)	Definition + Example	4M	4M	
6		Sample class diagram example , explanation of each i) attributes and operations ii) Qualified association iii) multiplicity iv) association end name v) generalization and inheritance	2*5	10M	10 M
7	a)	Definition of Multiple Inheritance **1.** Disjoint **2.** Overlapping Deligation(workarounds)	1M 2M 2M 5M	10M	10M
8		Class diagram with 3 generalization, multiplicity, association names	10M	10M	10M

1. a) What is object orientation? Explain in brief the stages involved in OO methodology. (08M)
    Software is organized as a collection of discrete objects that incorporate both State and behavior.
    OO methodology Stages
    The process for OO development and graphical notation for representing OO concepts consists of building a model of an application and then adding details to it during design.
    (5 different stages and sub types with explanation each carries 1 M)
    The methodology has the following stages: 6*1=6M

➢ System conception : Software development begins with business analysis or users conceiving an application and formulating tentative requirements

➢ Analysis: The analyst must work with the requestor to understand the problem, because problem statements are rarely complete or correct.
    The analysis model is a precise abstraction of what the desired system must do, not how it will be done.
    It should not contain implementation decisions.
    ➢ Domain Model
    ➢ Application Model

➢ System design: The development teams devise a high – level strategy – the system architecture for solving the application problem.
    They also establish policies that will serve as a default for the subsequent, more detailed portions of design.
    The system designer must decide what performance characteristics to optimize, choose a strategy of attacking the problem and make tentative resource allocations.

➢ Class design : The class designer adds details to the analysis model in accordance with the system design strategy.
    The focus of class design is the data structures and algorithms needed to implement each class.

➢ Implementation: Implementers translate the classes and relationships developed during class design into particular programming language, database or hardware.
    During implementation, it is important to follow good software engineering practice so that traceability to the design is apparent and so that the system remains flexible and extensible.
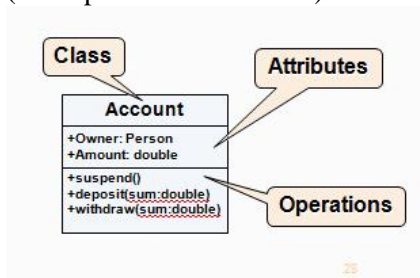
    b) Define Class Model with example.
    (Definition of Class model)   1M
    Class Model for the objects in the system & their relationships.
    (Example of Class Model)   1M

2. a) Elaborate on the major themes that are well supported in object oriented technology? (10M)
   (six points with explanation with diagrams each carries 1.5 M)   1.5*6=9M
    (Examples carries 1 M)   1M

➢ Abstraction: focus on essential aspects of an application while ignoring details.

➢ Encapsulation :(Information Hiding) separates the external aspects of an object , that are accessible to other objects from internal implementation details.

➢ Combining data and behaviour

➢ Sharing (reuse)

➢ Emphasis on the essence of an object: OO technology stresses what an object is, rather than how it is used.

➢ Synergy: Identity, classification, polymorphism, and inheritance characterize OO languages. Use all together.

3 a)
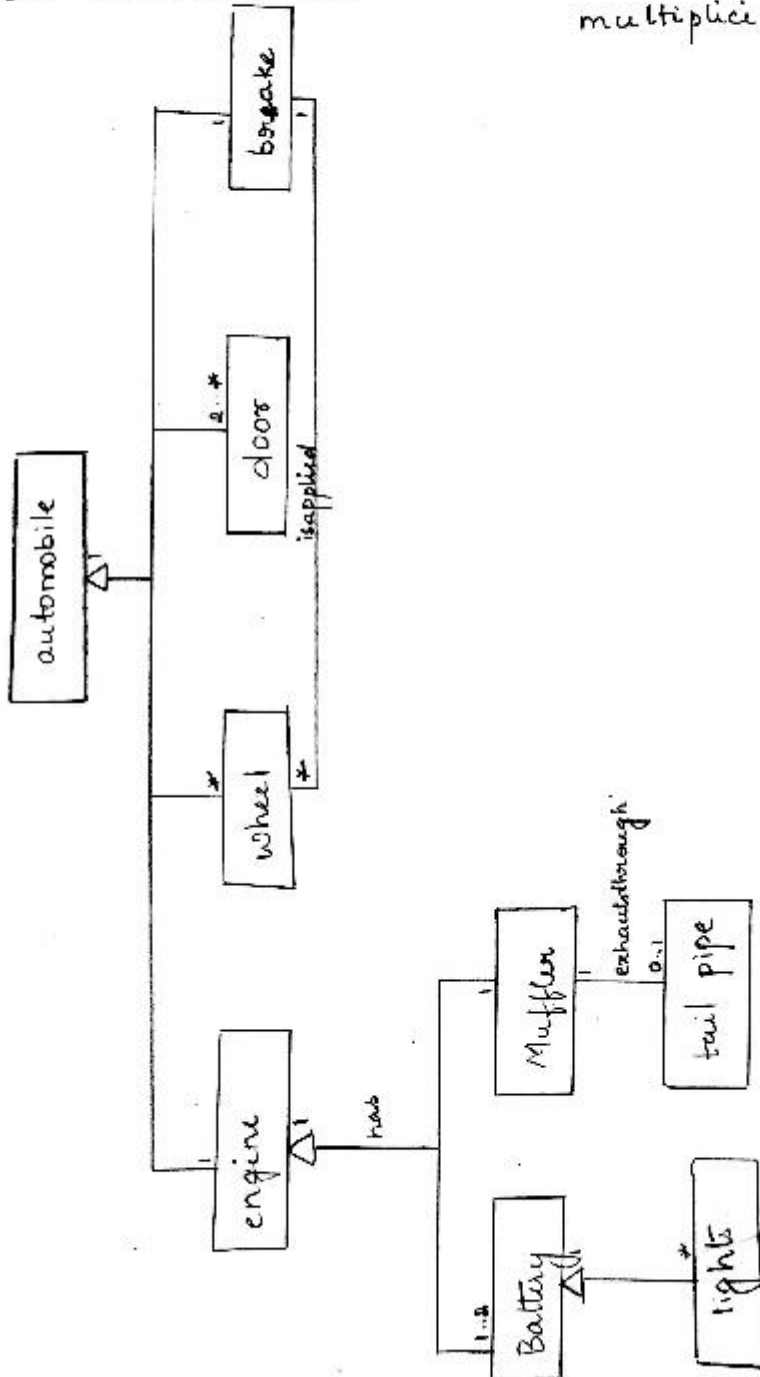
Scanned by CamScanner

4 a) Define Model? Explain the purpose of modeling.   (06M)

(Definition of Model)   2 M

A model is an abstraction, before building any system a prototype may be developed.

The main purpose of model is for understanding of the system.

Designer builds different kinds of models for various purposes before constructing things.

A model is a simplification of reality.

A model may provide

- o blueprints of a system
- o Organization of the system
- o Dynamic of the system

For example car , airplane, blueprints of machine parts, Plan for house construction etc., Models serve many purposes

Designers build many kinds of models for various purposes before constructing things.

Models serve several purposes –

(Explain any 4 Purpose each carries 1M)   1*4=4M

➢ Testing a physical entity before building (simulation)

➢ Communication with customer

➢ Visualization

➢ Reduction of complexity

➢ Better understanding of the problem

4 b) Define the ordering bags and Sequence with example.   4M

(Definition of ordering ,bags, Sequence carries 1 M each)   1*3=3M

(Examples carries 1M)   1M

- The ordering is an inherent part of the association. If objects indicate ordered set objects by

  writing   {ordered}   next to appropriate association end.

  Example: A workstation screen contains a number of overlapping windows. Each window on a screen occurs at most once. The windows have an explicit order so only the topmost window is visible.

Figure 3.15 Ordering the objects for an association end. Ordering sometimes occurs for "many" multiplicity.

- A sequence is an ordered collection of elements with duplicates allowed

- Bag: collection of elements with duplicates allowed.

- •Sequence: ordered collection of elements with duplicate allowed eg. An itinerary is a sequence of airports and the same airport can be visited more than once.

- Sequence is ordered bag allow duplicates, {ordered} and {sequence}

Figure 3.16 An example of a sequence. An itinerary may visit multiple airports, so you should use {sequence} and not {ordered}.

5. a) Explain Aggregation, Composition and Propagation of operation with example (6M)
(Aggregation, Composition and Propagation of operation definition and Examples each carries 1 M)

2*3=6M

Aggregation

✓ is a strong form of association in which an aggregate object is formed using other objects as parts
✓ An aggregate object is treated as a unit in many operations
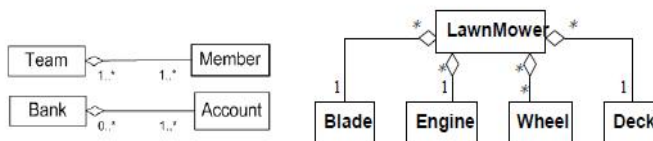
Composition

✓ Is a form of aggregation with two additional constraints
✓ Deletion of an assembly objects triggers deletions of all constituent objects

| Company | 1 ◆━━━━* | Division | 1 ◆━━━━* | Department |

1
| WorksFor
*
| Person |
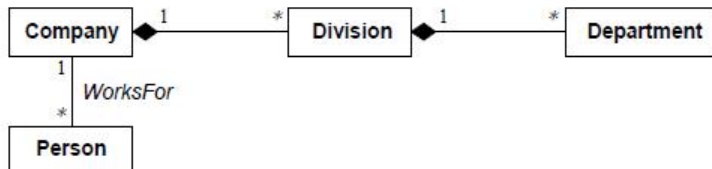
Propagation of Operations

✓ Propagation is a powerful way for specifying continuum of behavior
✓ Operations are propagated to parts via composition relations

| Person | Owns | Document | copy → | Paragraph | copy → | Character |
| | 1   * | copy | 1   * | copy | 1   * | copy |
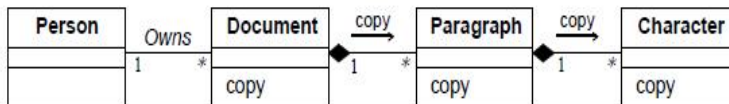
5 b) Define N ary Association? Explain how to avoid   with example.   4M

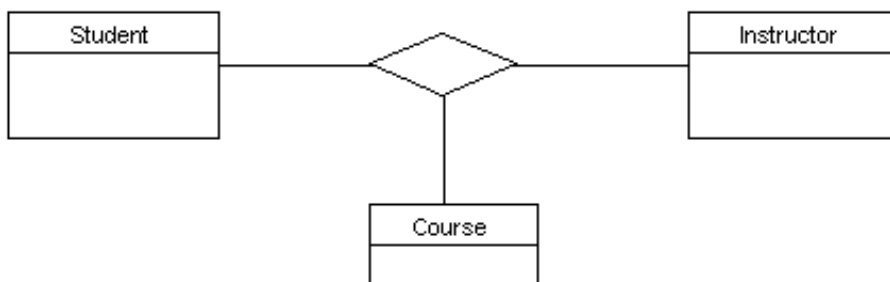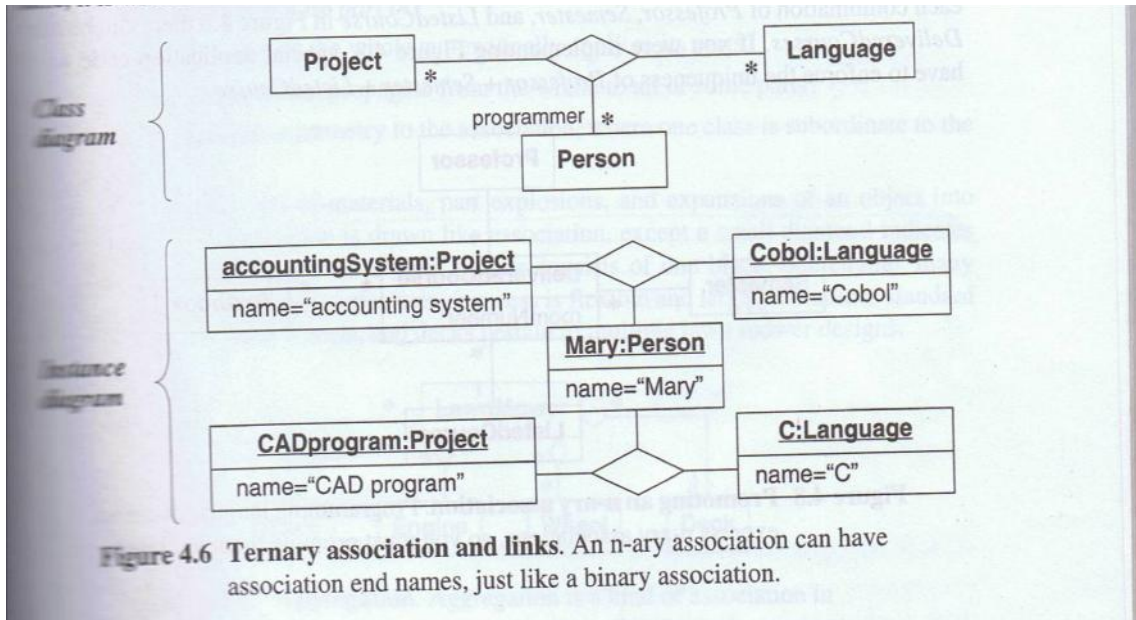(Definition + Example)   2 M

An n-ary association is like a common (binary) association, except that more than two association roles involved in it.
- The UML symbol for n-ary associations is a diamond with lines connecting to related classes. If the association has a name, it is written in italics next to the diamond.

| Student | ◇ | Instructor |

| Course |

Figure 4.6 Ternary association and links. An n-ary association can have association end names, just like a binary association.

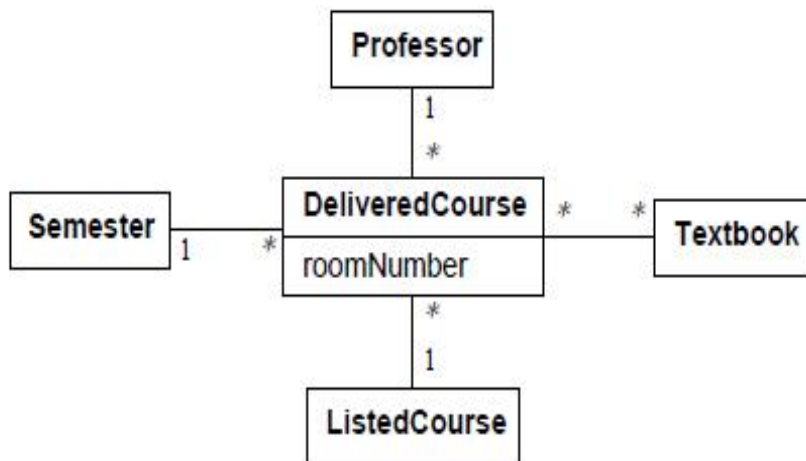# N – ary associations can be viewed as association classes

✓ Cannot be traversed like binary associations
✓ OCL does not support notation for traversing n-ary associations
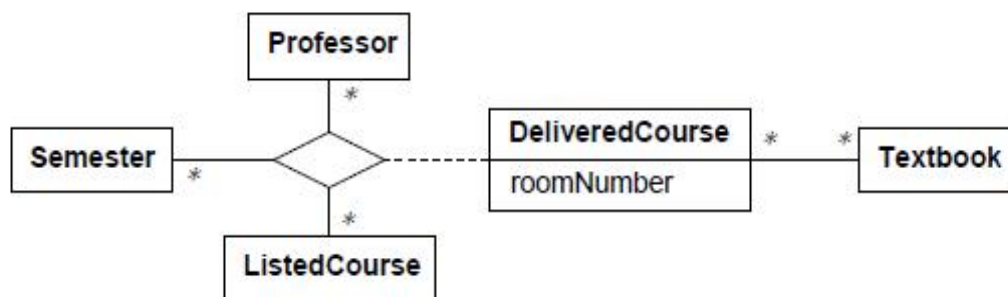
(How to avoid) 2M

N-ary associations as classes

✓ Typical programming language cannot express n-ary associations

Drawback: Programming languages don't support N-ary association. (2M)

6. With the help of a sample class model explain the following i) attributes and operations ii) Qualified association iii) multiplicity iv) association end name v) generalization and inheritance.   10M
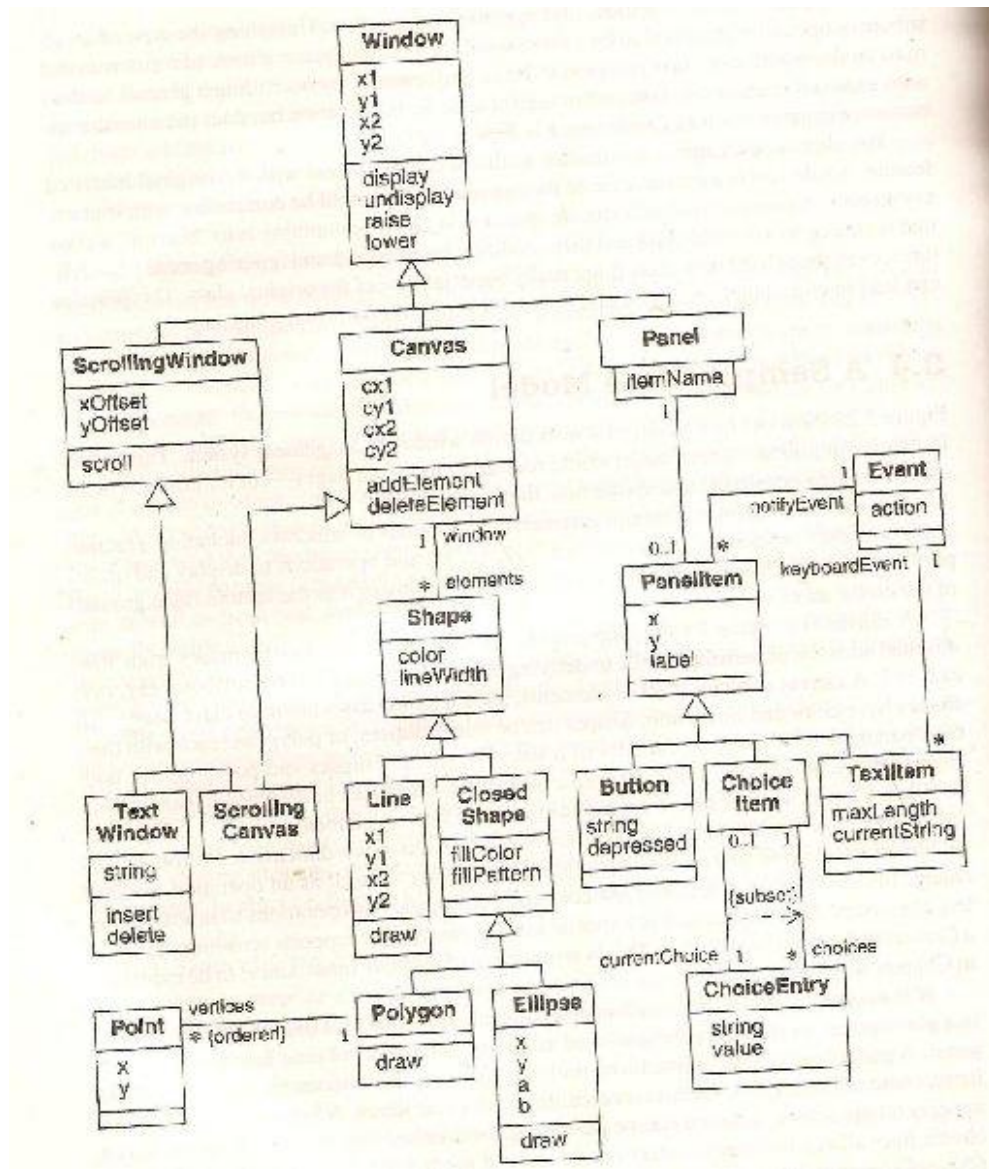
7. Define Multiple Inheritance and Explain different kinds of Multiple inheritance and workarounds. 10M
(Definition of Multiple Inheritance carry 1M) 1M

Multiple Inheritance

✓ permits a class to have more than one superclass

```
                          ┌──────────────┐
              ┌──────────▷│   Employee   │◁──────────┐
              │           └──────────────┘           │
  employmentStatus                          managerialStatus
     ┌────────┴────────┐              ┌──────────┴──────────┐
┌──────────────────┐ ┌──────────────────┐ ┌──────────┐ ┌─────────────────────┐
│ FullTimeEmployee │ │ PartTimeEmployee │ │ Manager  │ │ IndividualContributor │
└──────────────────┘ └──────────────────┘ └──────────┘ └─────────────────────┘
         △                                                        △
         │                                                        │
         └──────────────────────┬─────────────────────────────────┘
                    ┌─────────────────────────────┐
                    │ FullTimeIndividualContributor │
                    └─────────────────────────────┘
```
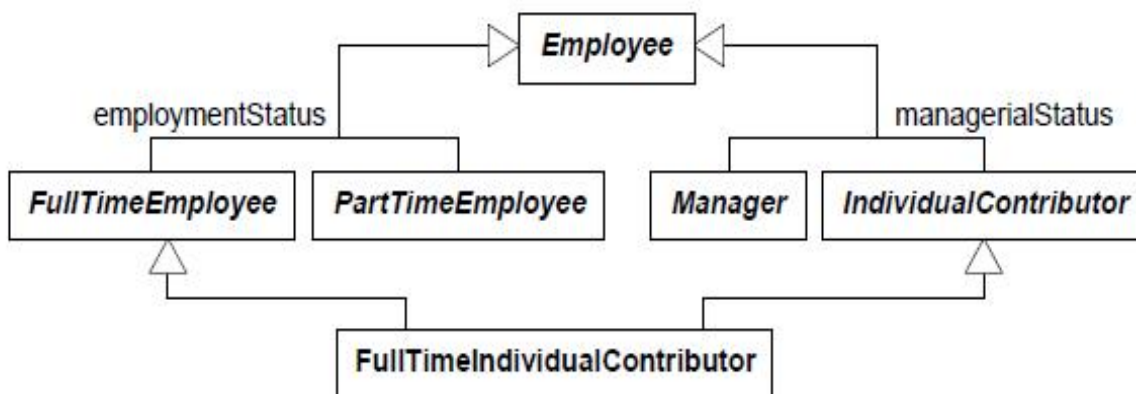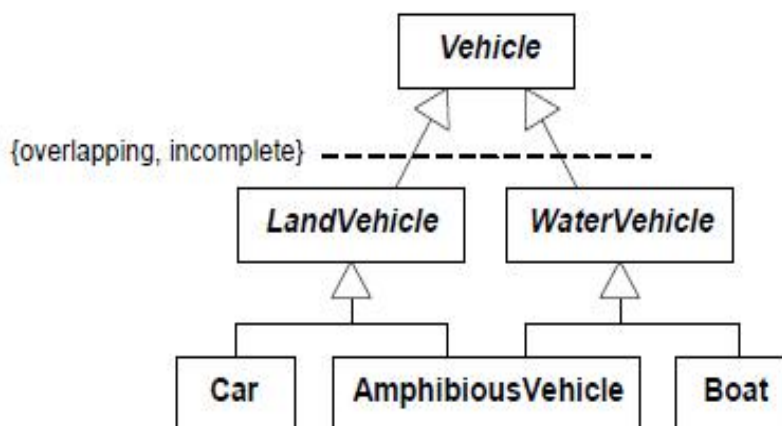
Multiple inheritance from sets of disjoint classes (partitions)

Multiple Inheritance

✓ Conflicts between features of different inheritance paths must be resolved

```
                    ┌──────────┐
                    │ Vehicle  │
                    └──────────┘
                      △      △
{overlapping, incomplete} ──/────────────
                    △              △
            ┌─────────────┐  ┌──────────────┐
            │ LandVehicle │  │ WaterVehicle │
            └─────────────┘  └──────────────┘
                  △                 △
          ┌───────┴──────┐   ┌──────┴──────┐
      ┌───────┐ ┌───────────────────┐ ┌────────┐
      │  Car  │ │ AmphibiousVehicle │ │  Boat  │
      └───────┘ └───────────────────┘ └────────┘
```

Multiple inheritance from overlapping classes

(ThreeKinds of Multiple inhertiance each carries 3 M)

Nested Generalization

✓ Preserves object identity, but duplicates declarations (and code)
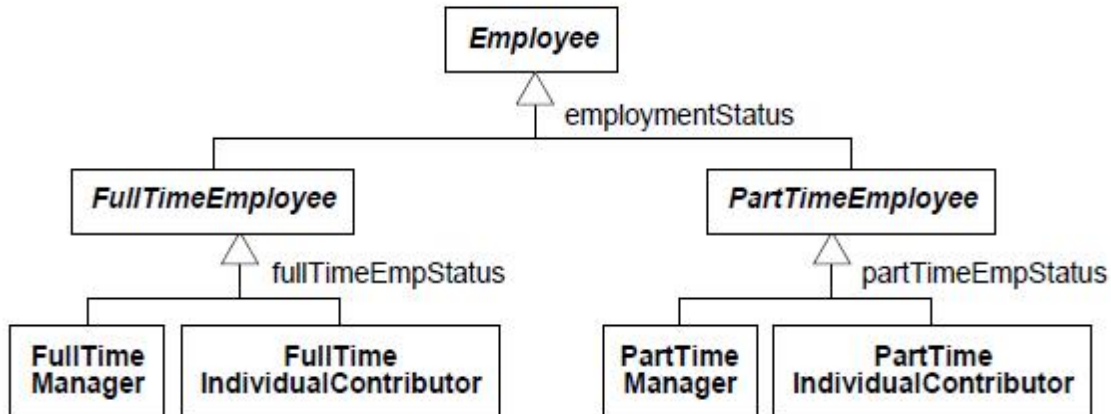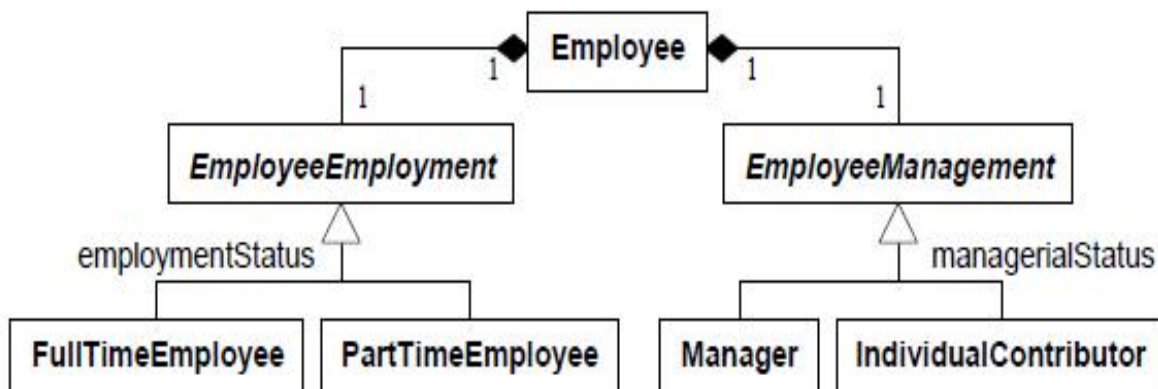
Delegation

✓ Replaces a unique object identity by an artificially created group of objects

✓ The composite object must catch operations and delegate to appropriate part

8)