SLIIT

Discover Your Future

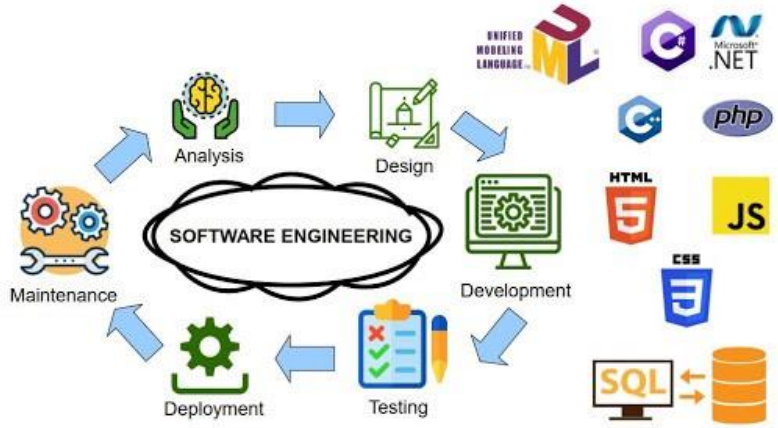
Software Engineering (IT2020)
2022

Lecture 1 – Introduction & Object Diagram

Outline

1. What is Software Engineering ?
2. Software Development Life Cycle and phases (Covered in previous semester)
3. UML Diagrams (Covered in previous semester)
4. Class Diagram (Covered in previous semester)
5. Object Diagrams

What is Software Engineering?



- IEEE Definition of Software Engineering:

The application of a systematic, disciplined, quantifiable approach for the development, operation, and maintenance of software.

Ref : IEEE Standard 610.12-1990, 1993.

- **Software engineering** is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

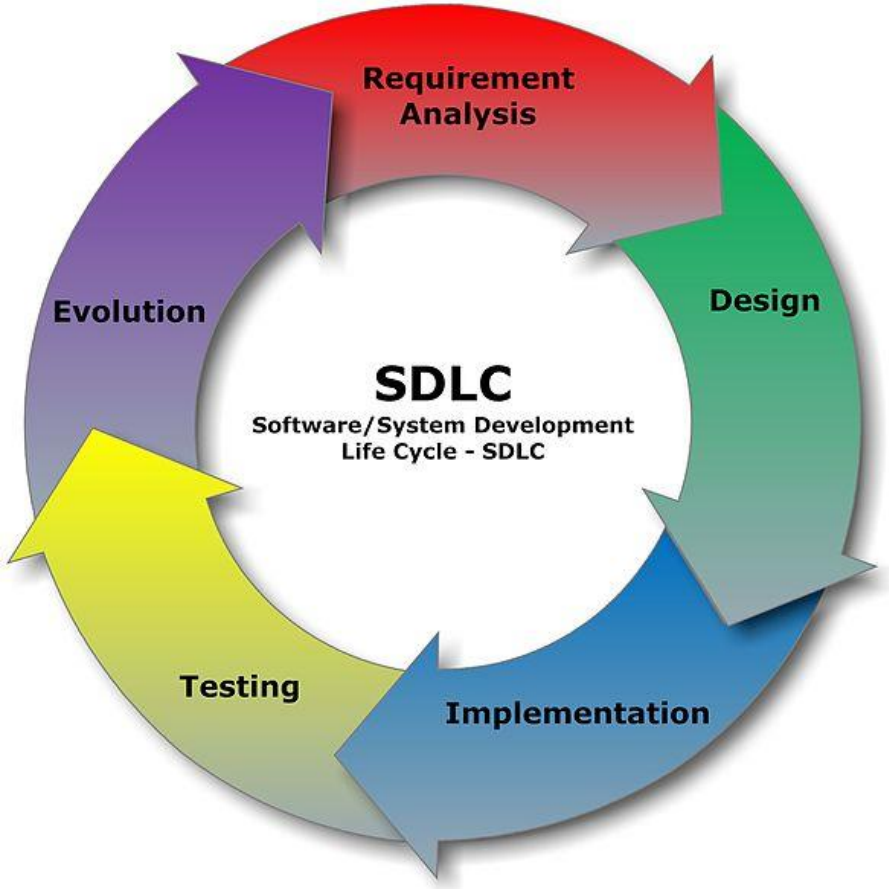
Software Development Process

- In Software Engineering, an Engineering process is followed to transform inputs into outputs/software products.

- The software development process consists of a set of activities and associated results that produce a Software.

Software Development Life Cycle

- Software Development Life Cycle (SDLC) is a framework that defines the phases/stages to be followed throughout the software development process.



Software Development Life Cycle

Phase 1 : Requirement Gathering and Analysis

Phase 2 : **Design**

Phase 3 : Implementation

Phase 4 : Testing

Phase 5 : Maintenance / Evolution

Design Phase

Software Design Methods

- **Function Oriented Software Design**
- **Object Oriented Software Design**

In SE Module, we are going to cover Object Oriented Software Design.

Object Oriented Design

- **Object Oriented Software Design:**
- Object-oriented design is the discipline of defining the objects and their interactions to solve a software problem.

- Object Oriented Concepts are the base for the object-oriented design.

Object Oriented Software Design Cont…

Design Model Types

- **Structural Models**
- **Dynamic Models**

Modeling Languages

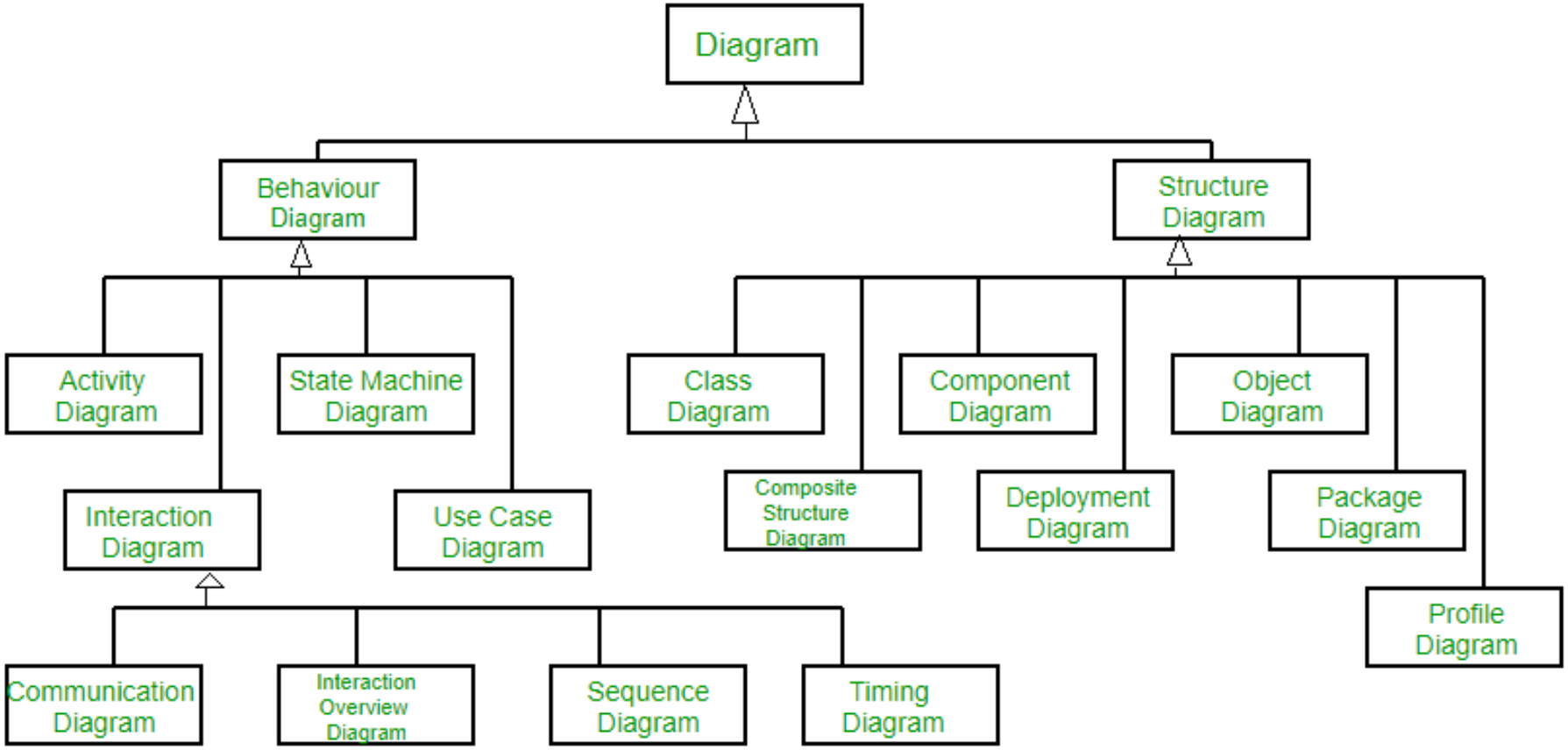
A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules.

Unified Modeling Language (UML)

What Is the UML?

UML is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

UML Diagram Structure



Object Oriented Software Design Cont…

In SE module we are going to learn following UML Diagrams.

- **Class Diagram- Completed in OOC**
- **Object Diagram**
- **Sequence and Communication Diagrams –(Interaction Diagrams)**
- **State Diagram**
- **Component and Deployment Diagrams –(Physical Diagrams)**

Class Diagram Revision

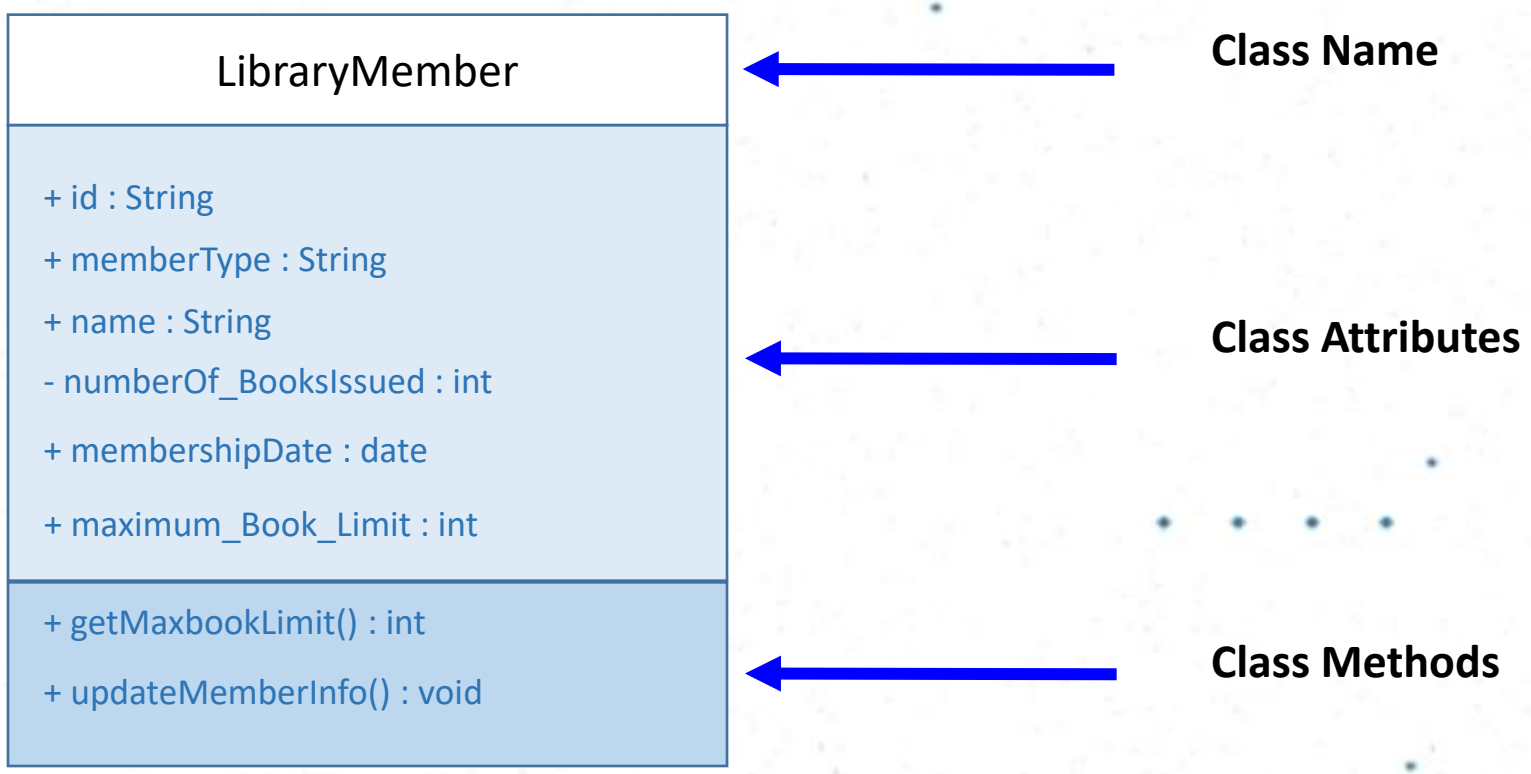
How to discover classes?

Noun/ Verb Analysis

- Through Noun/Verb Analysis, we can identify objects in our problem statement by looking for nouns and noun phrases.
- Each of these can be underlined and becomes a candidate for an object in our solution.
- Then write **Class Responsibility and Collaboration (CRC)** Cards for final set of classes.

Class Diagram

Class Structure



Class Structure Cont…

Class Attributes

- Attributes of a class can be fully specified as below.

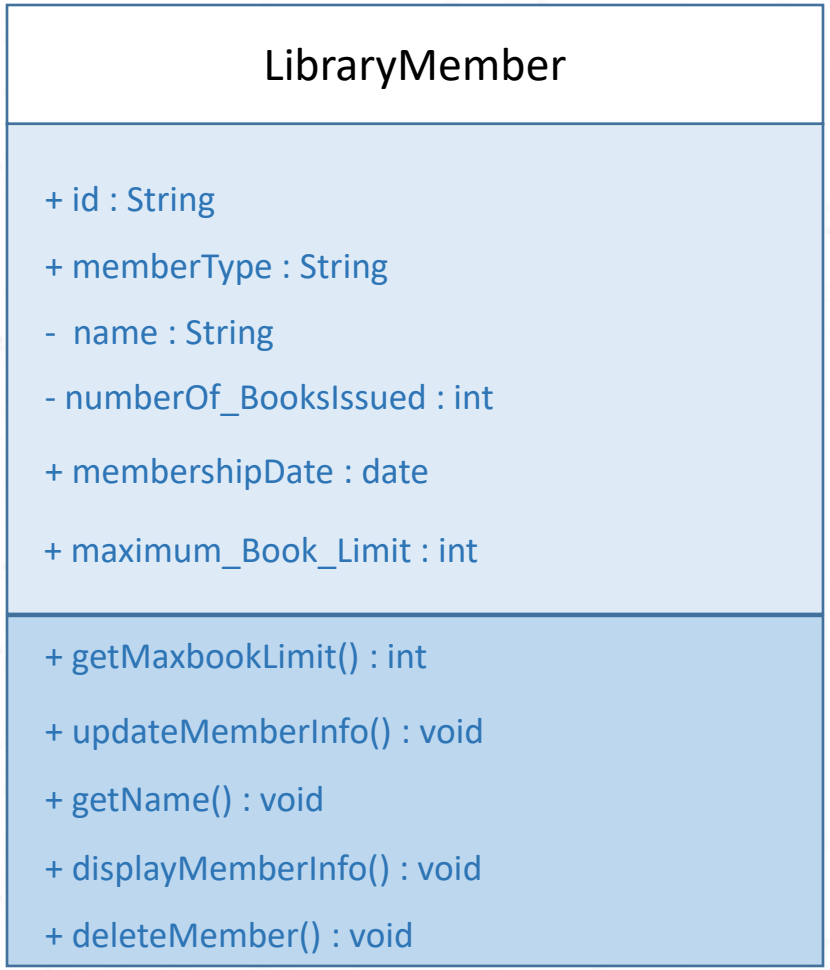
visibility **name** multiplicity : type = initial value {property}

Class Methods

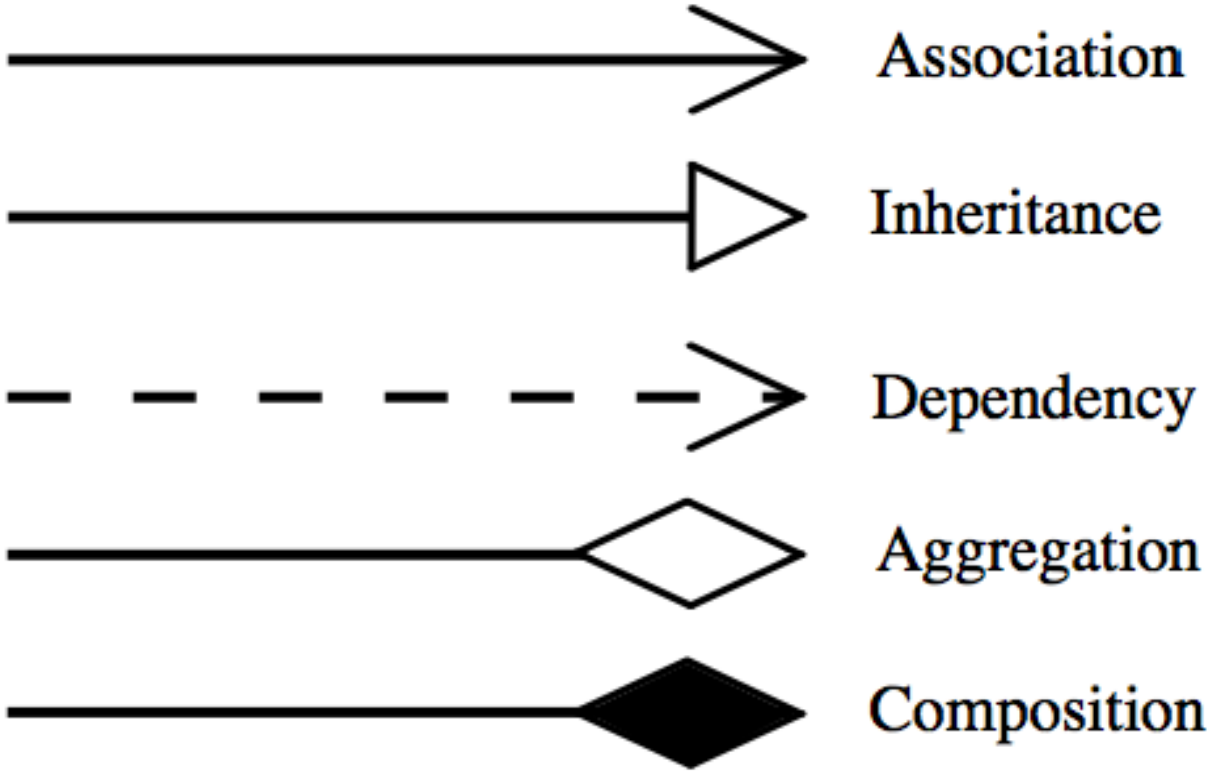
- Methods of a class can be fully specified as below.

name (parameters) : type

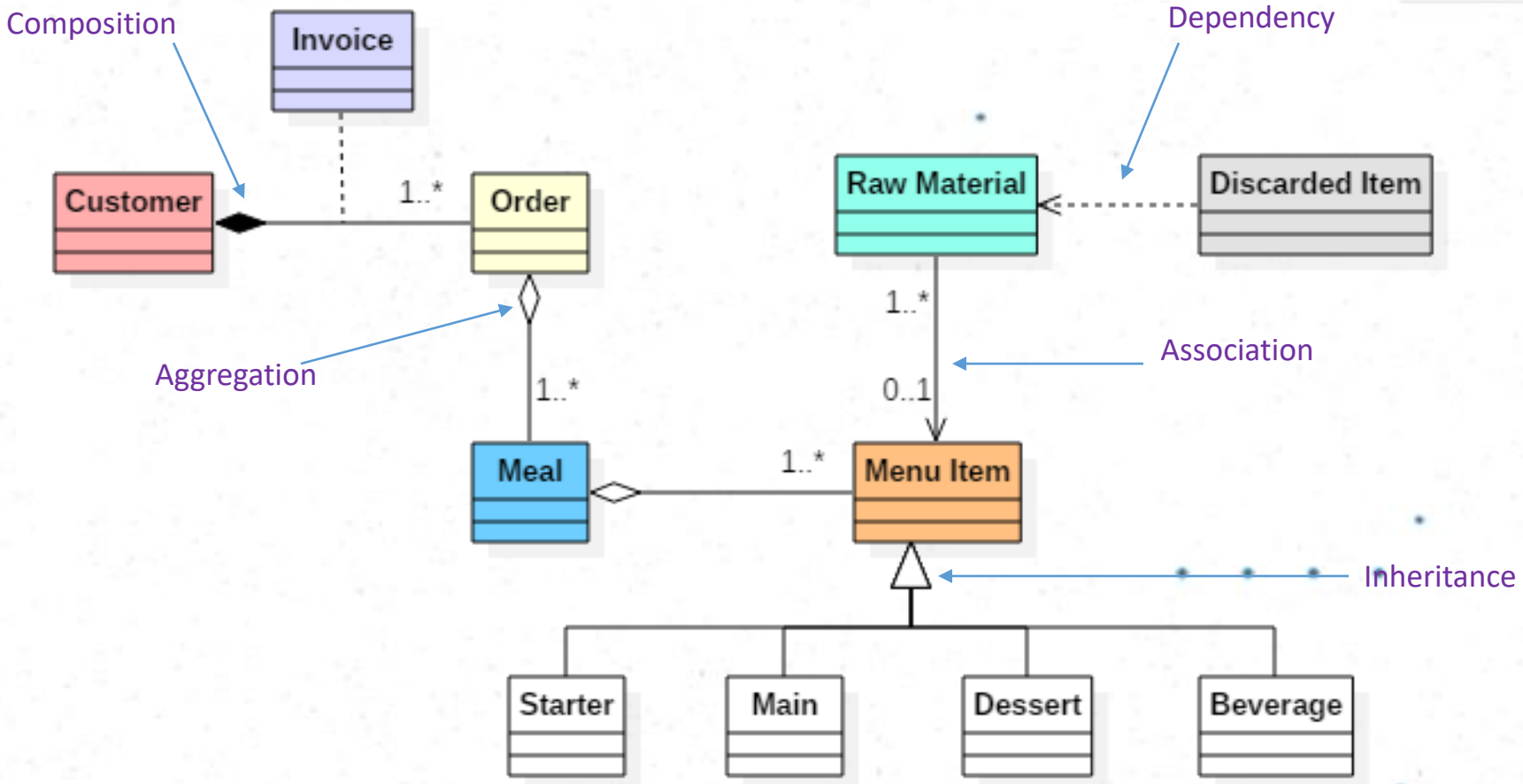
Class Structure Example



Class Relationships



Class Relationships Cont.



Exercise 1

As discuss in OOC, draw a class diagram for SLIIT Library Management System.

Object Diagram

Object Diagram

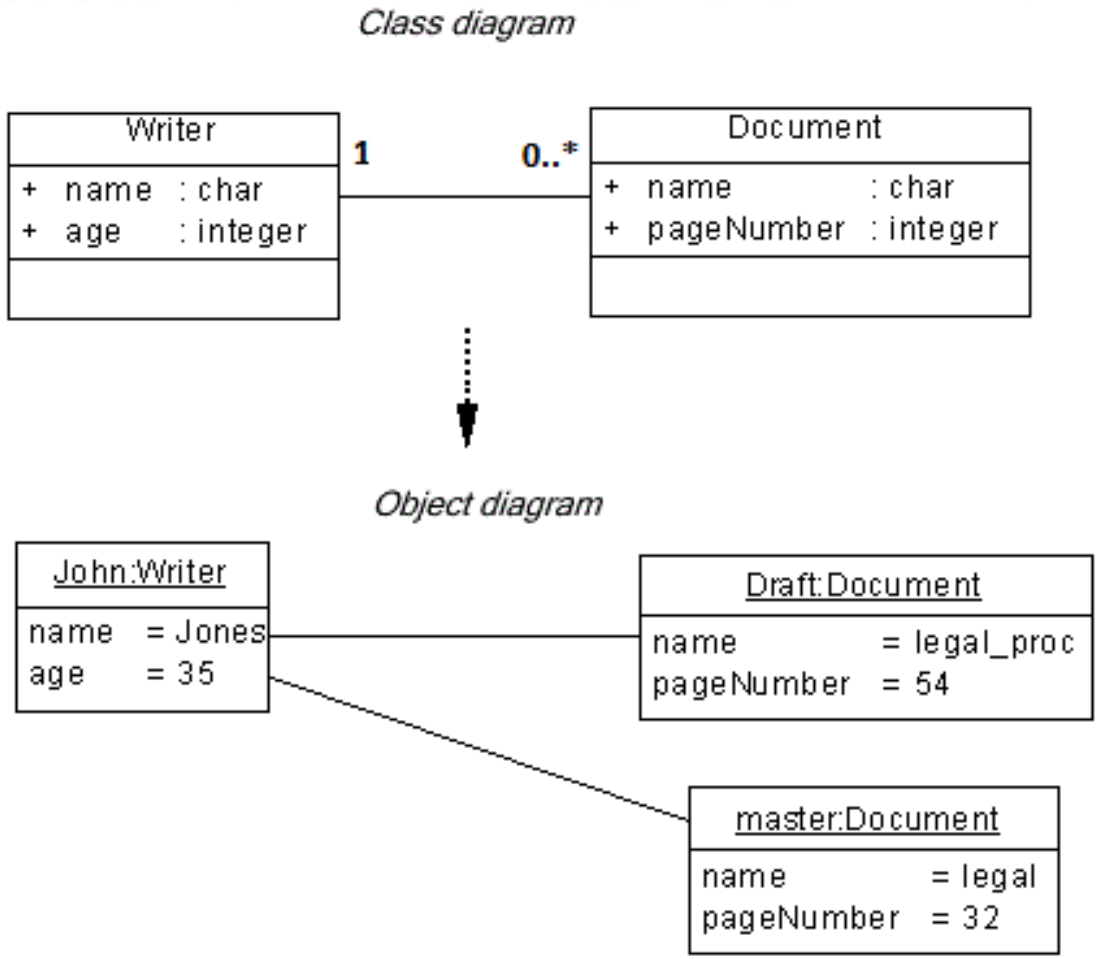
UML Specification:

"An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time."

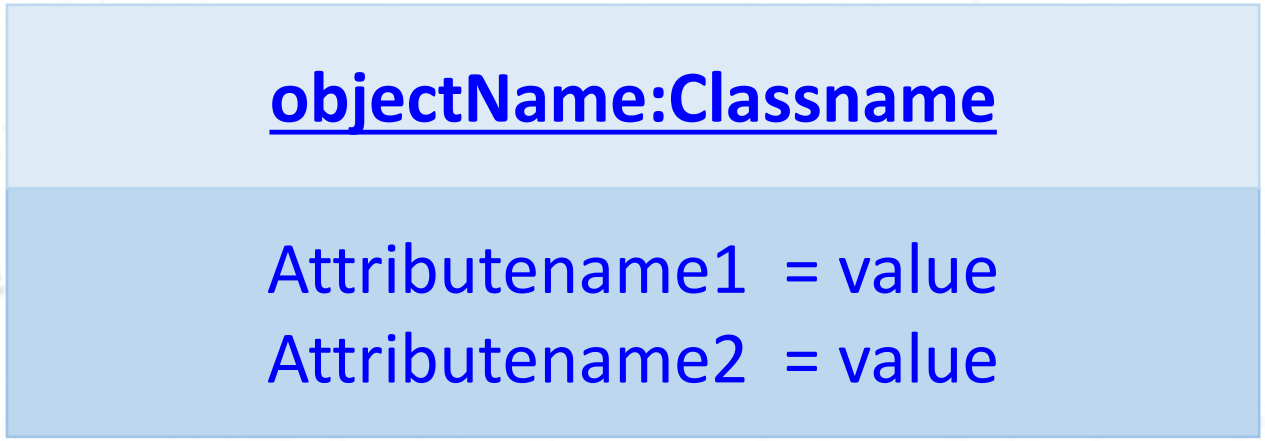
Object Diagram

- Object diagrams are derived from class diagrams, so object diagrams are dependent upon class diagrams.
- Both Class and Object diagrams are meant to visualize the structure of a system. Hence categorize under **Structural** diagram.
- Object diagrams represent an instance of a class diagram.
- The attributes identified by the class now have **values** associated with it.
- The purpose is to capture the static view of a system at a particular moment.

Object Diagram Example



Object Notation



- Top compartment contains object name and class name.
- Bottom compartment contains list of attribute names and values assigned. Attribute types are not shown.
- No need to show the operations (they are the same for all objects of a class)

Different Notation Types

Named Object :

Object name and the class name both should be there.

Objectname : Classname

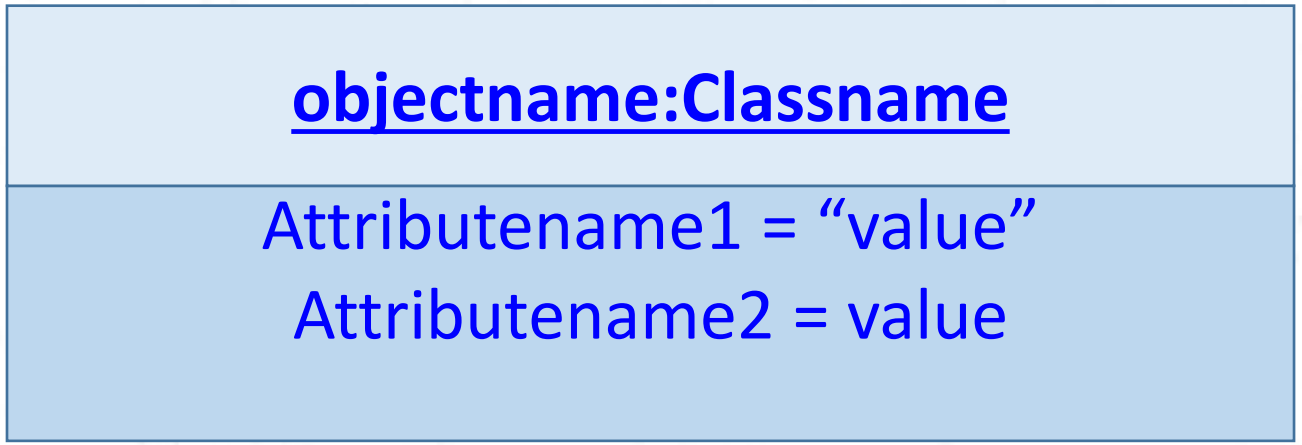
Anonymous Object :

The name of the object may be omitted (optional), but the colon should be kept with the class name.

:Classname

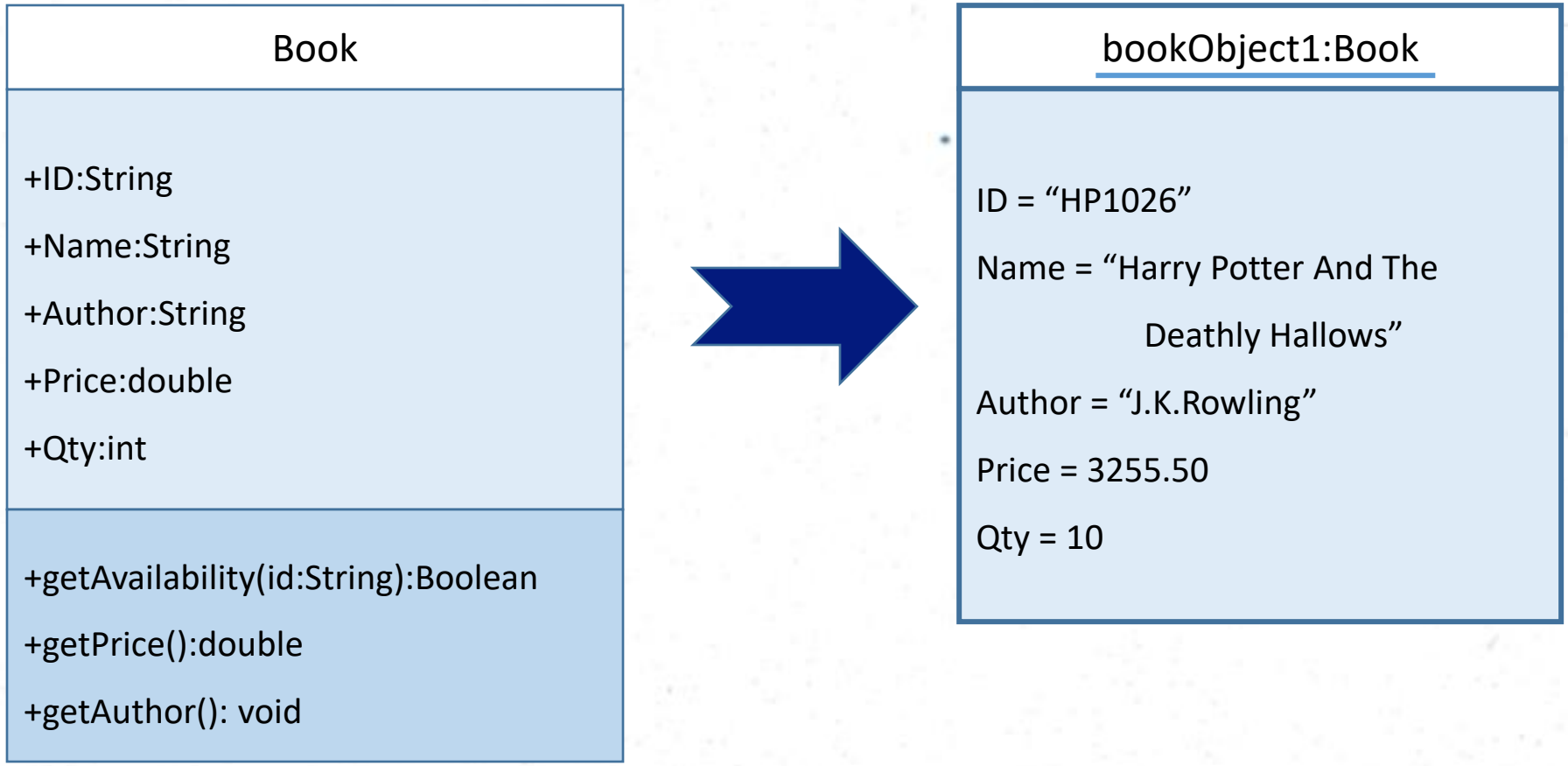
Shorter form of Notation

Object With Attributes :



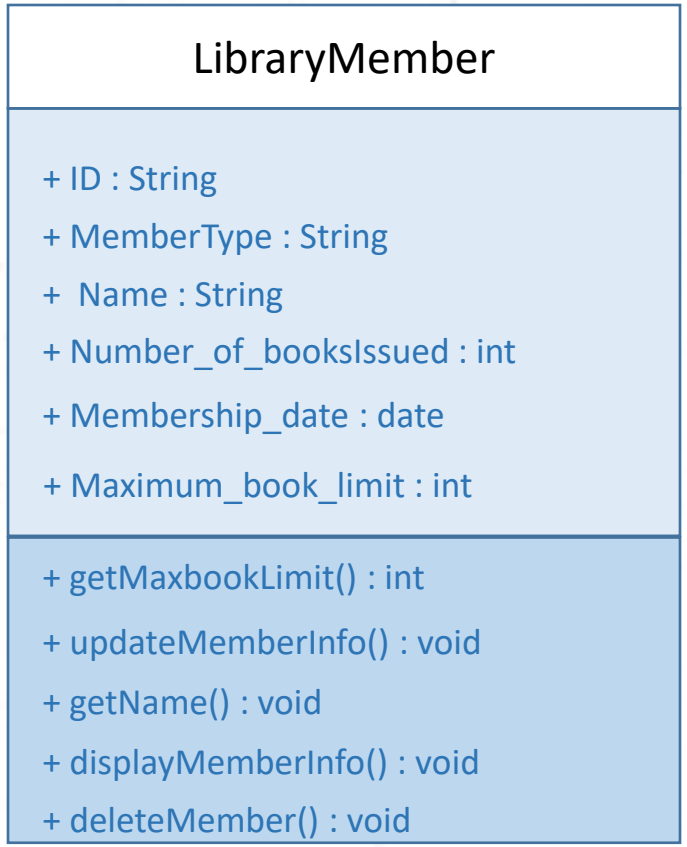
Note: Double quotes (“ ”)are used for String values.

Sample Object Diagram in UML



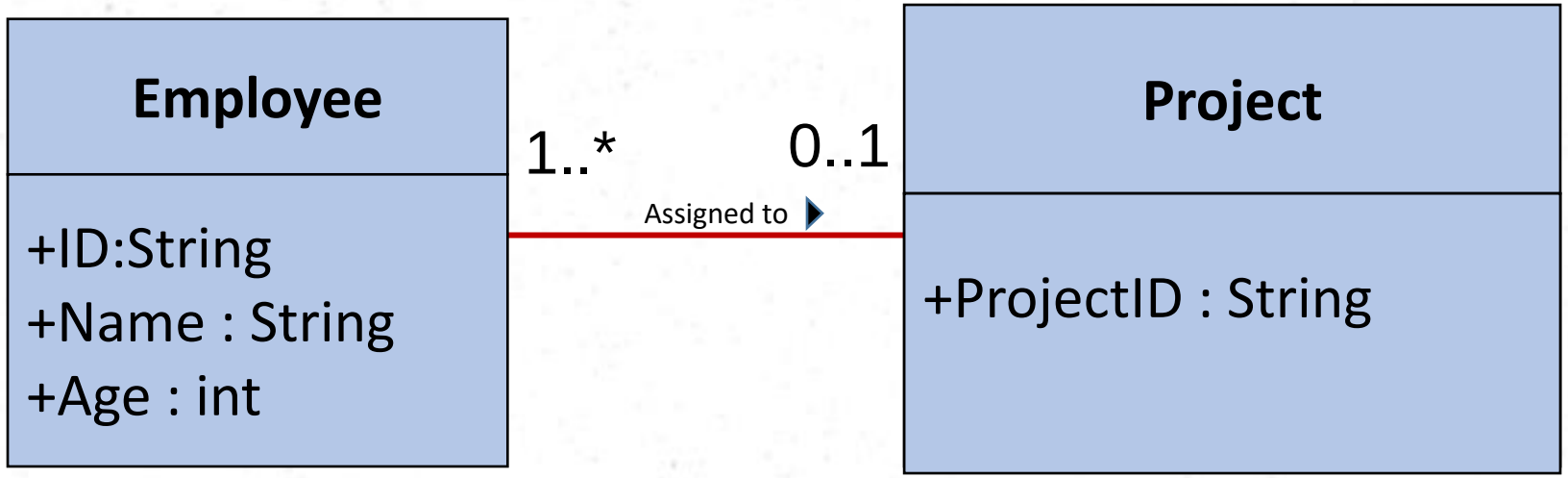
Exercise 2

Assume that you are the Library Member. Draw your Library Member object.



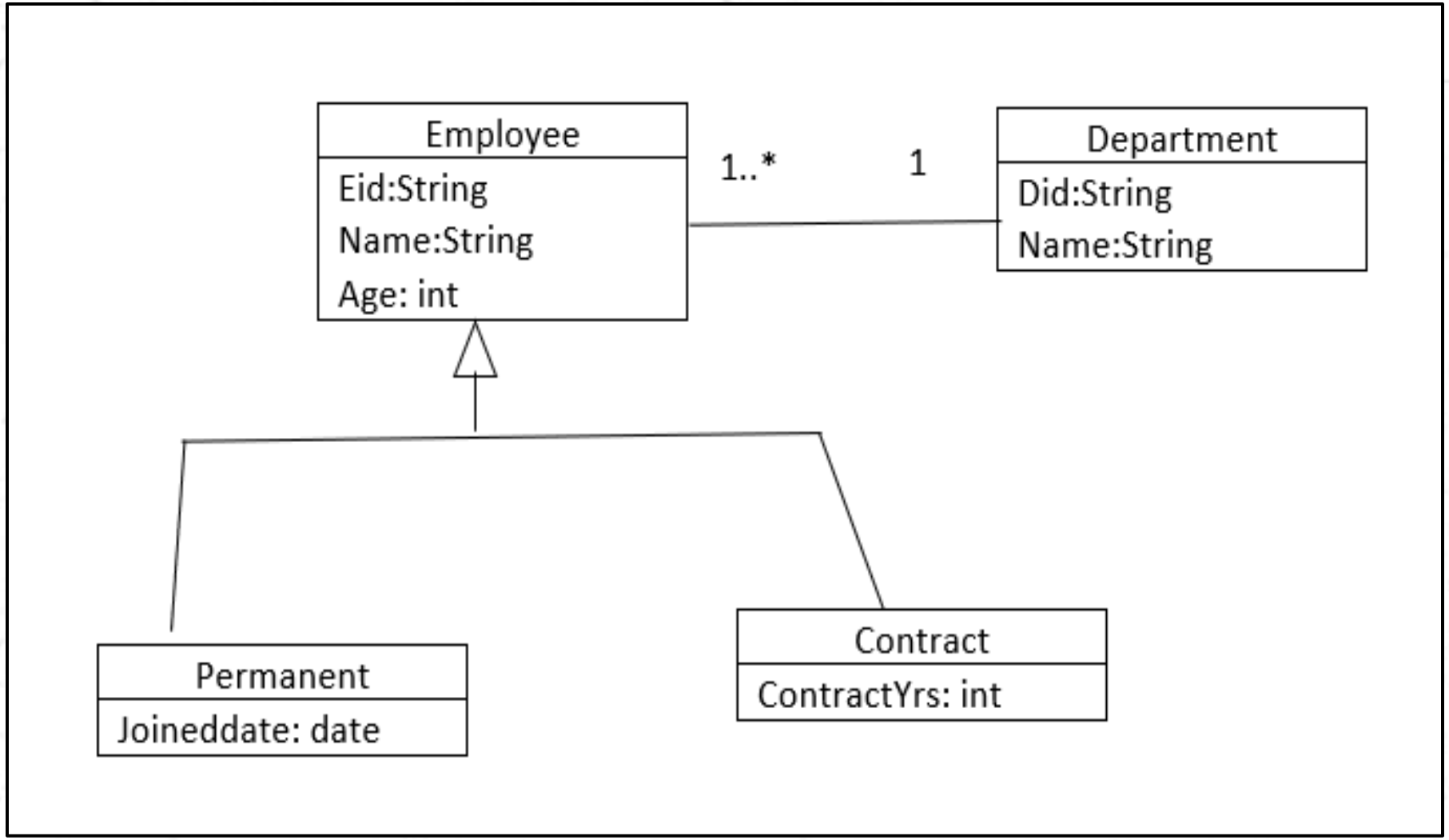
Exercise 3

Draw an Object Diagram



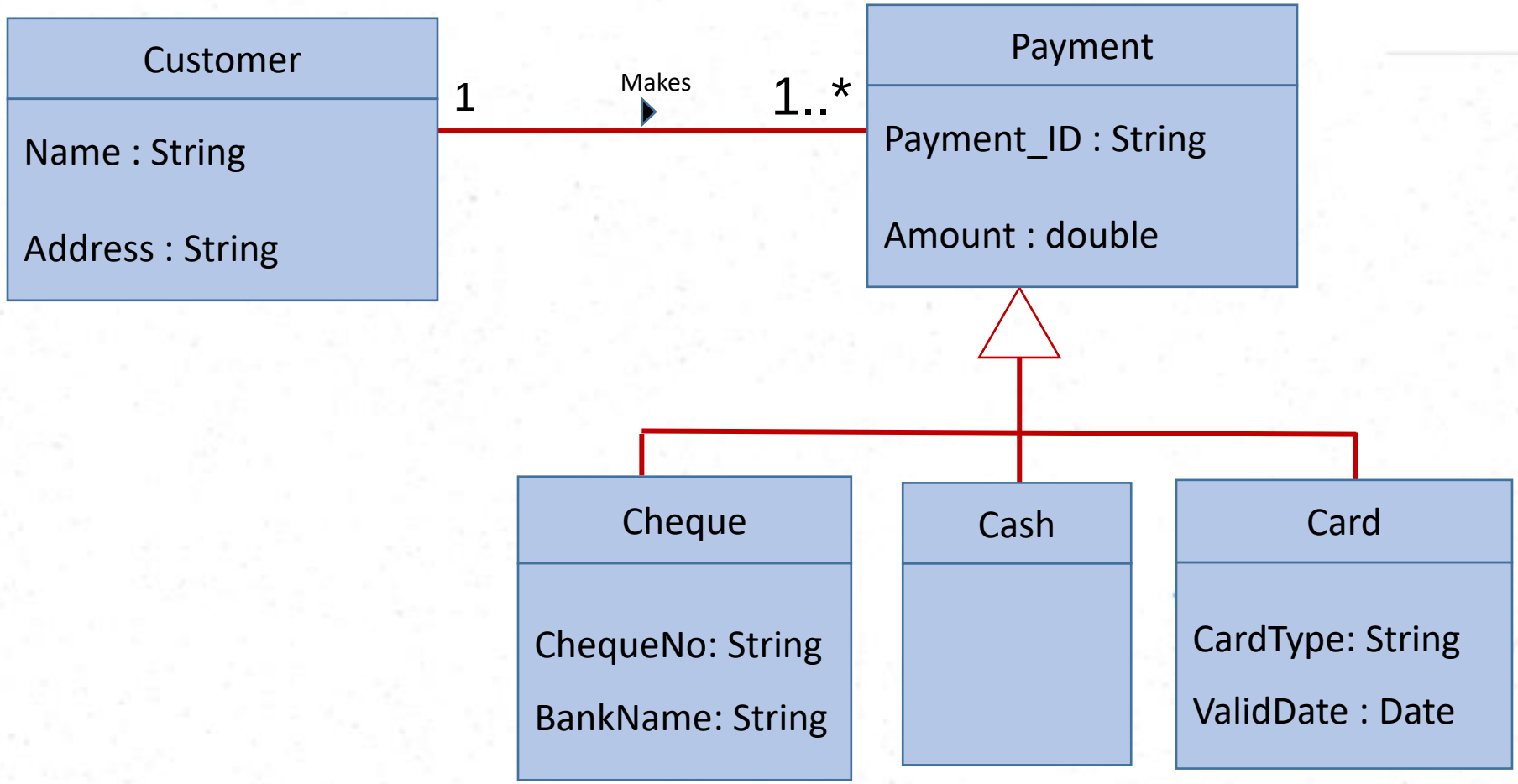
Exercise 4

Draw an Object Diagram for the given partial class diagram.



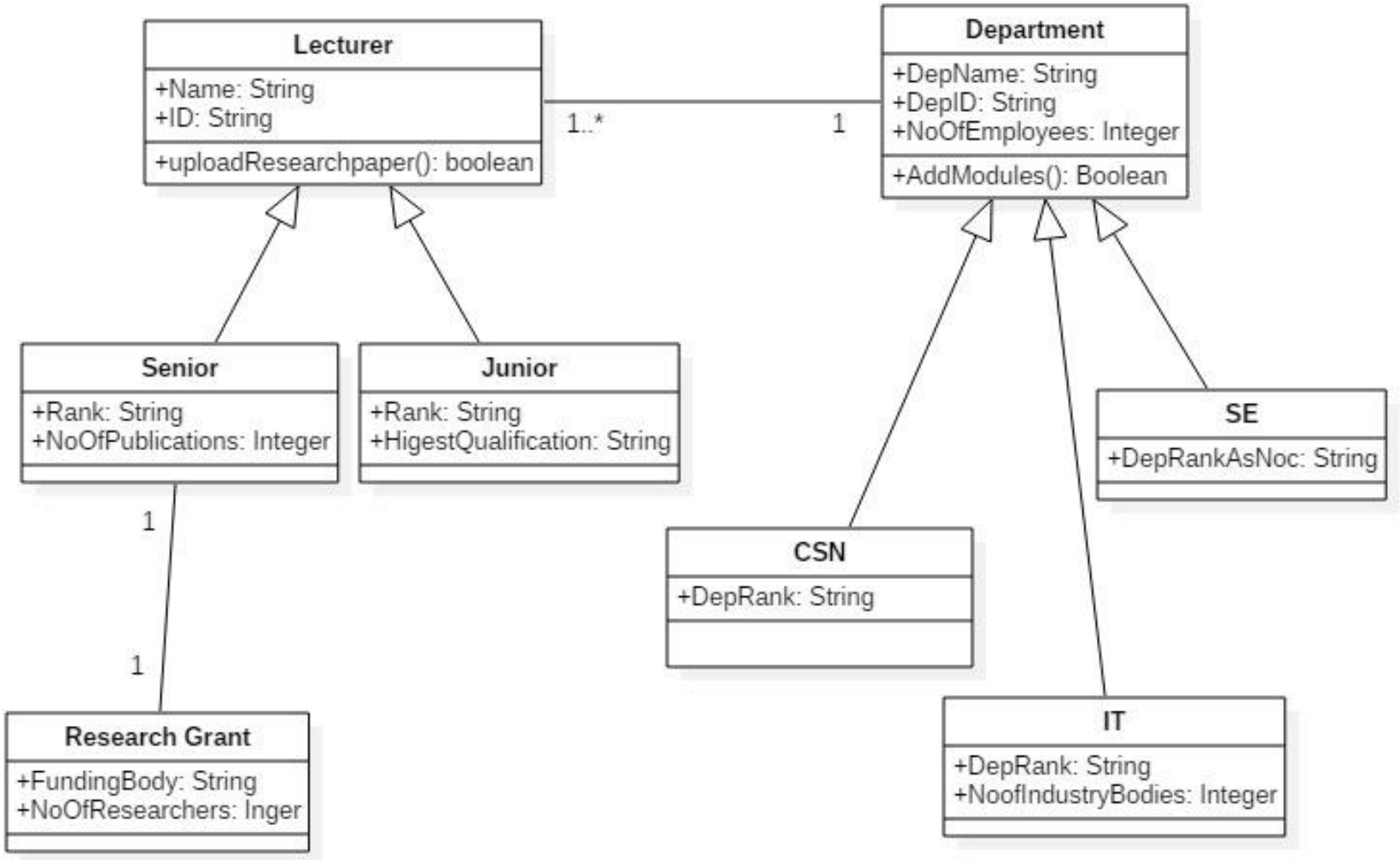
Exercise 5

Draw an Object Diagram for the given partial class diagram.



Exercise 06 – Self-study Question

Draw an Object Diagram for the given partial class diagram.



References

- IEEE Standard 610.12-1990, 1993.
- Software Engineering, I.Sommerville, 10th ed. , Pearson Education. (p. 21)
- Grady Booch, eta (2008), Object Oriented Analysis and Design with Applications 3rd Edition, pg 44,52)

Thank you.