TUTORIAL 6

Class Diagram

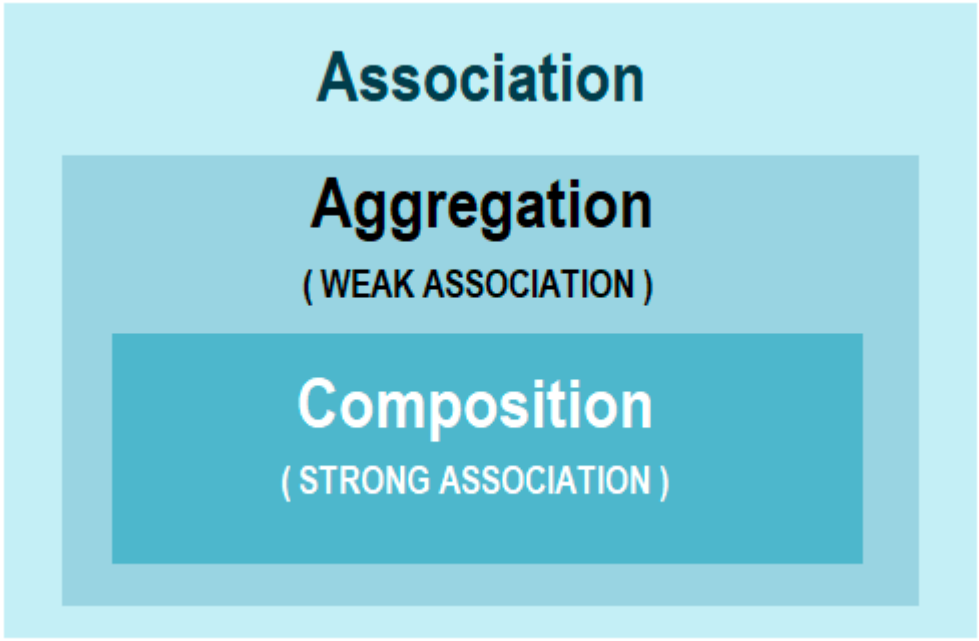
Tasks:

- 1. Draw Unified Modelling Language (UML) **class diagram**
  - Structural diagram
  - Static, does not move
  - Just relate class to class together using appropriate relationships

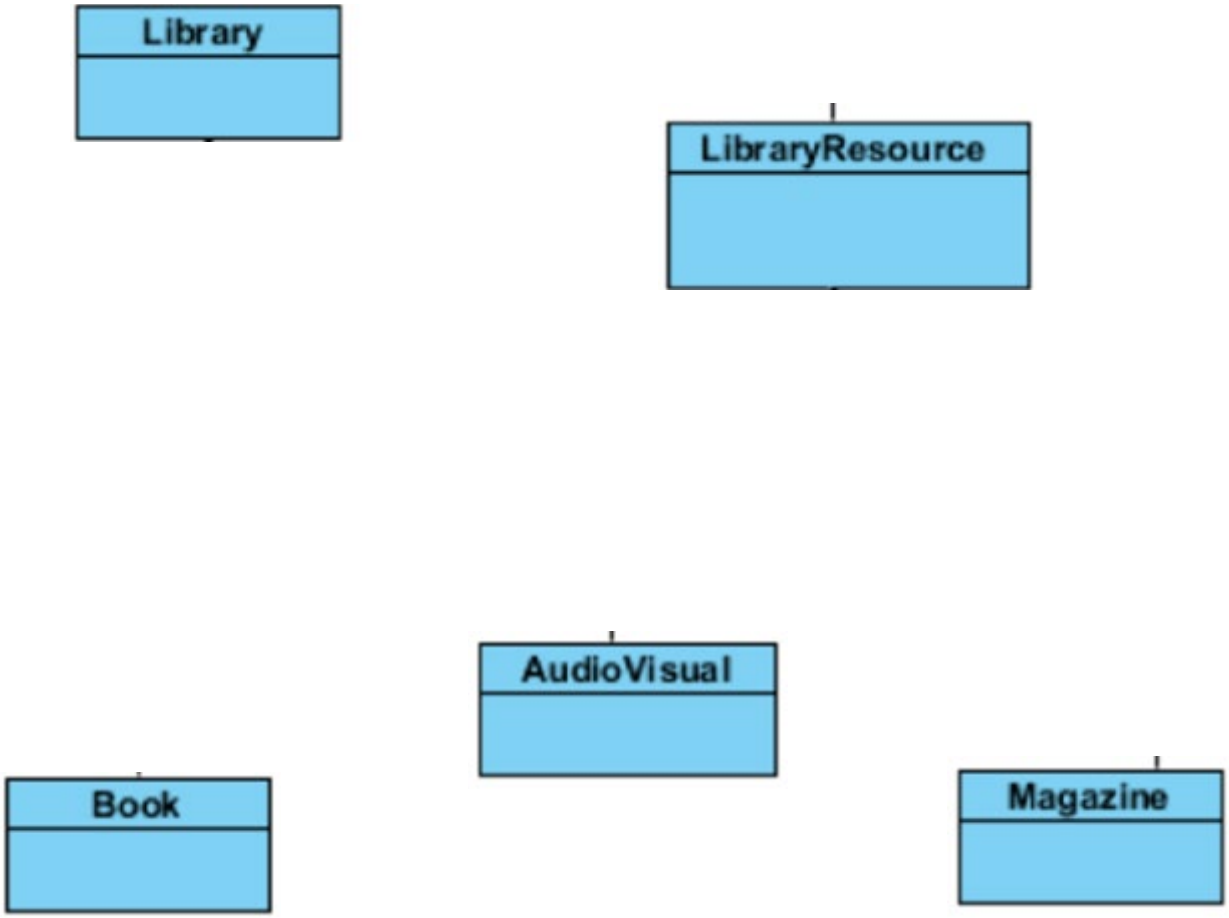
Q1

- Given the following set of classes, draw a Class Diagram to show the appropriate relationship between them. Add multiplicity, role name and association name, if necessary:
  - Library, LibraryResource, Book, AudioVisual, Magazine
  - Driver, Car, Wheel, Engine
  - Plane, City, Passenger, FlightTicket
  - Company, Person, Department, Job
  - Product, Inventory, ItemStock, Catalog, Order, OrderLineItem, Manufacturer

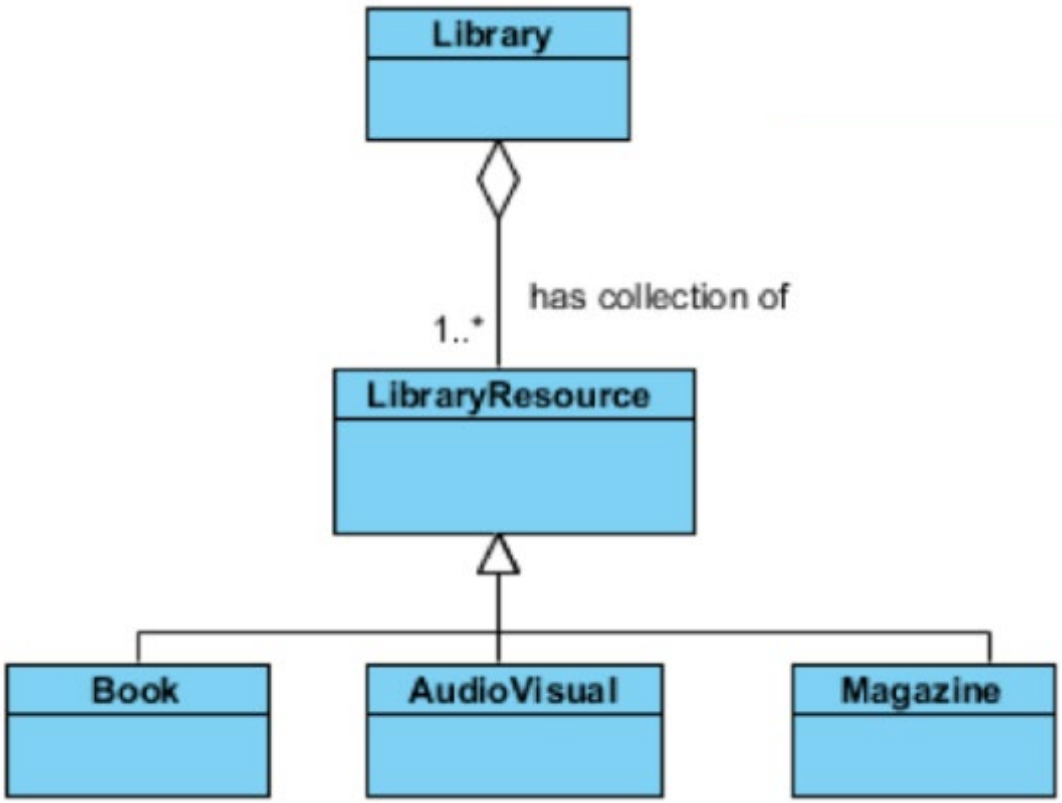
Association, Composition and Aggregation in Java



Q1A: Library, LibraryResource, Book, AudioVisual, Magazine

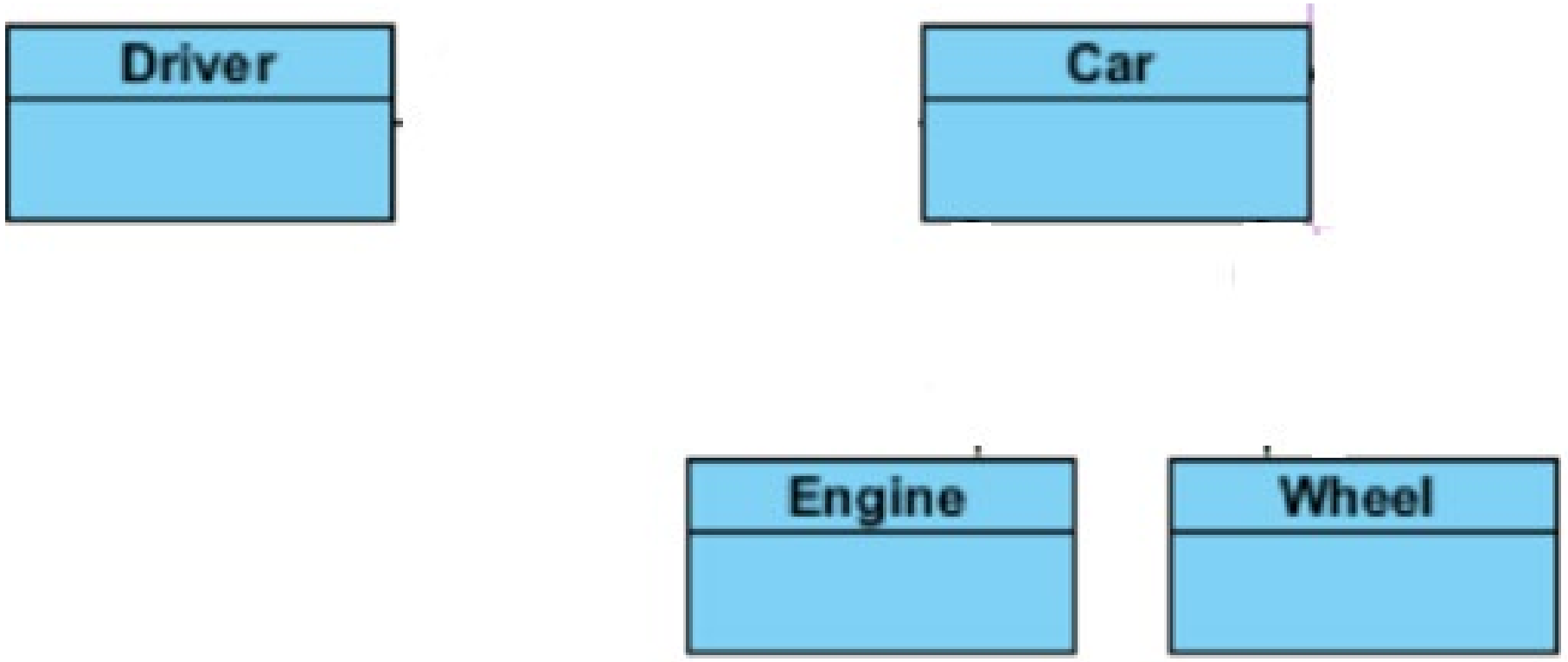

Library
LibraryResource
AudioVisual
Book
Magazine

A1.a)

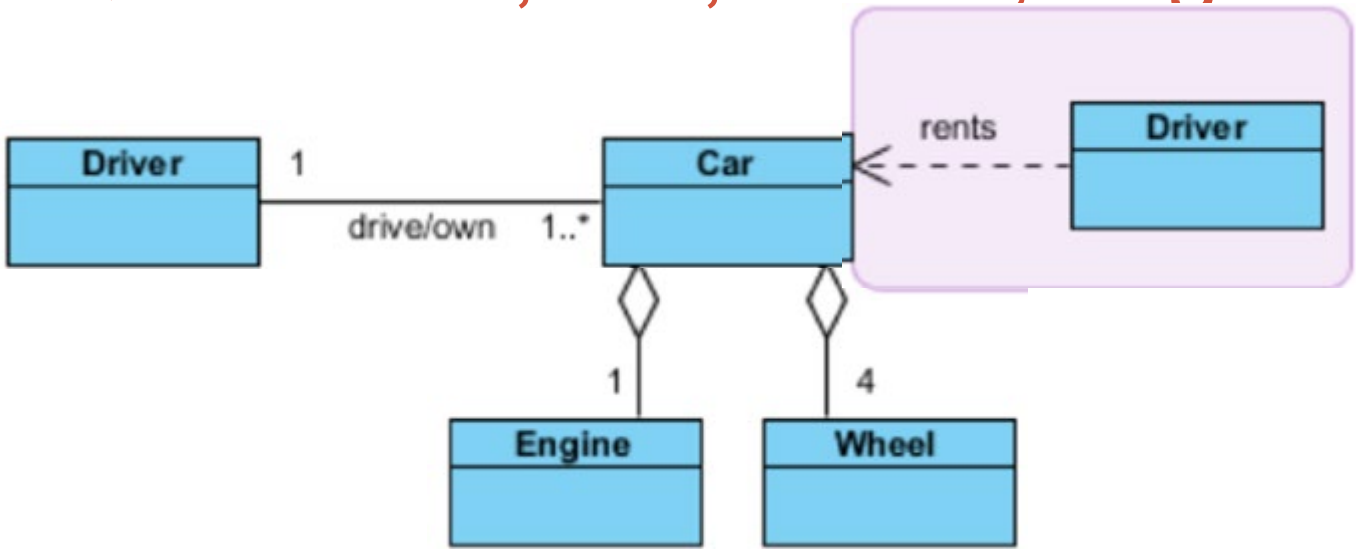


- Aggregation between Library and LibraryResource
- LibraryResource can be generalization or aggregation with Book, Magazine and AV.

Q1B: Driver, Car, Wheel, Engine

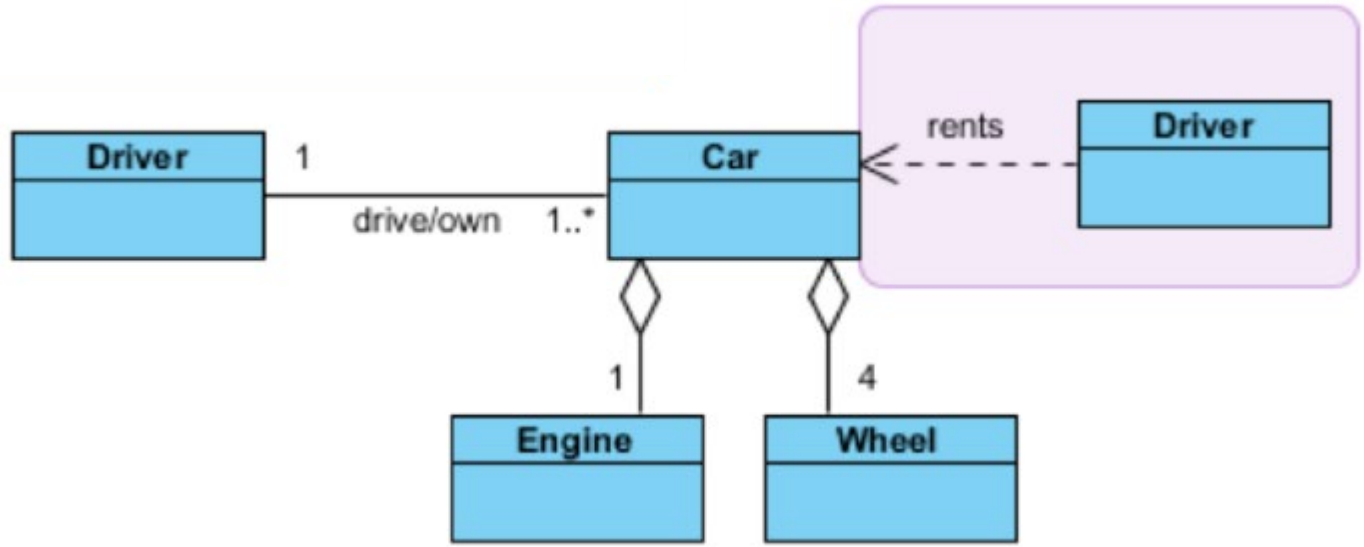


Q1B: Driver, Car, Wheel, Engine



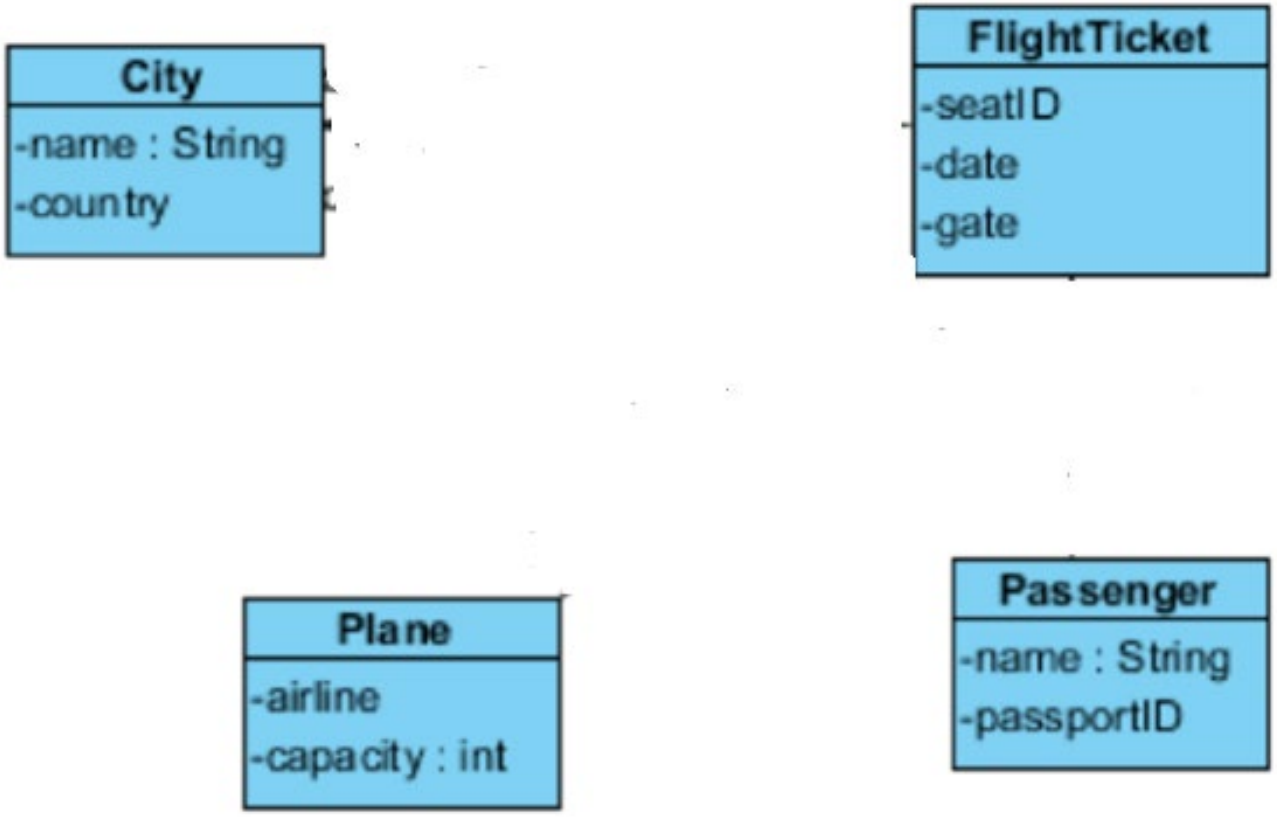
- if it is rental car

A1.b)

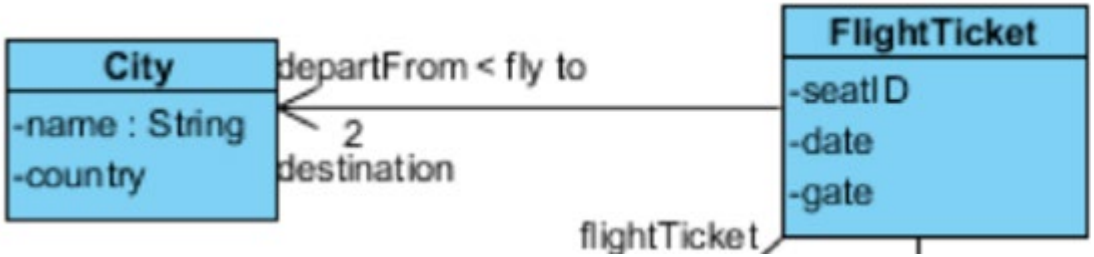


- Driver and Car can show dependency if it is rental car

Q1C: Plane, City, Passenger, FlightTicket

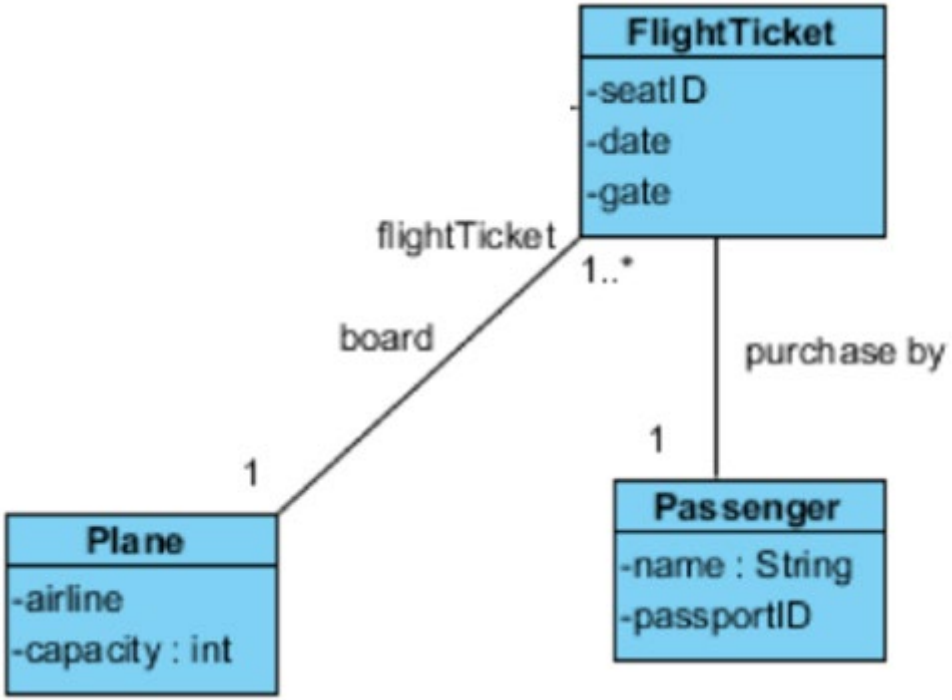


Q1C: Plane, City, Passenger, FlightTicket (City, FlightTicket)



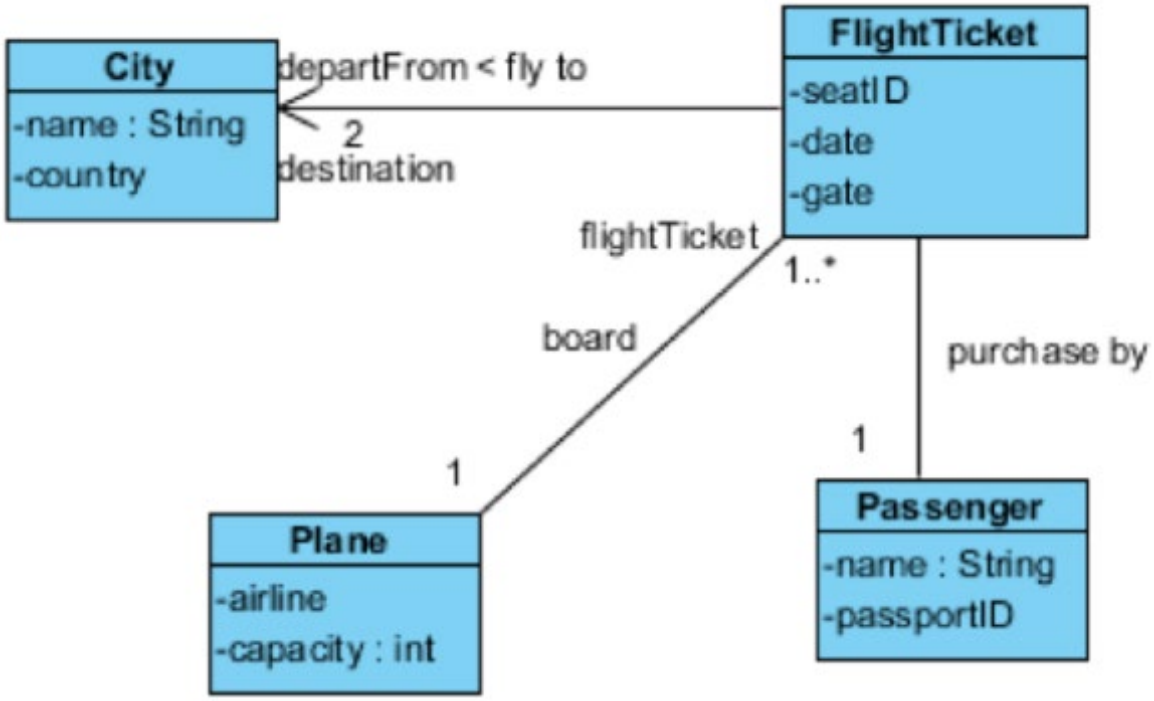
A role name identifies an end of an association and ideally describes the role played by objects in the association.

Plane, City, Passenger, FlightTicket (Plane, Passenger, FlightTicket)



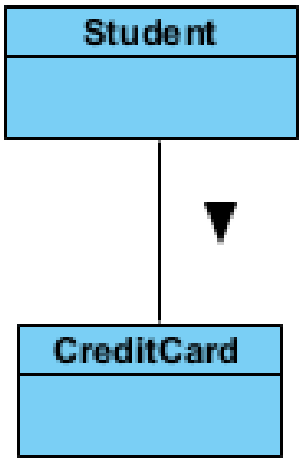
- FlightTicket can also be association class between Plane and Passenger.

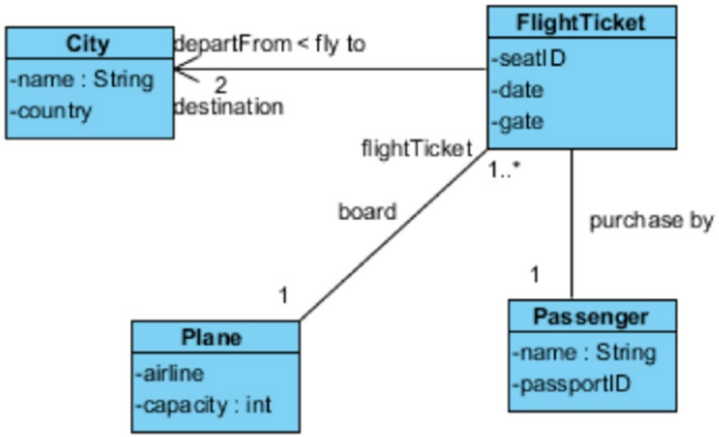
A1.c)

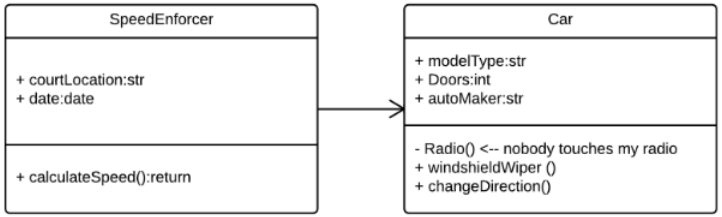


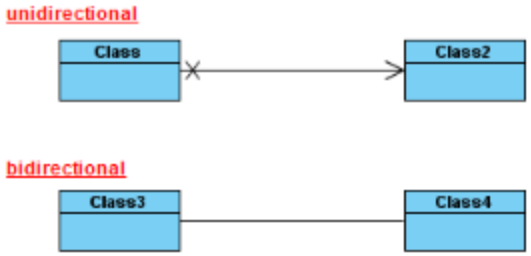
- FlightTicket can also be association class between Plane and Passenger.

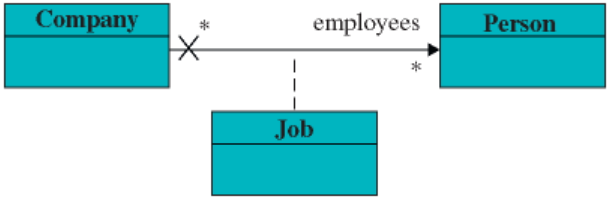
Directed Association （Uni-directional Association） in different UML tools

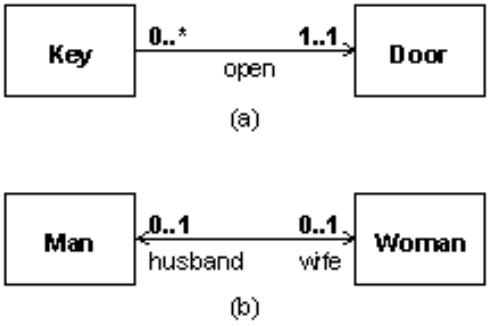




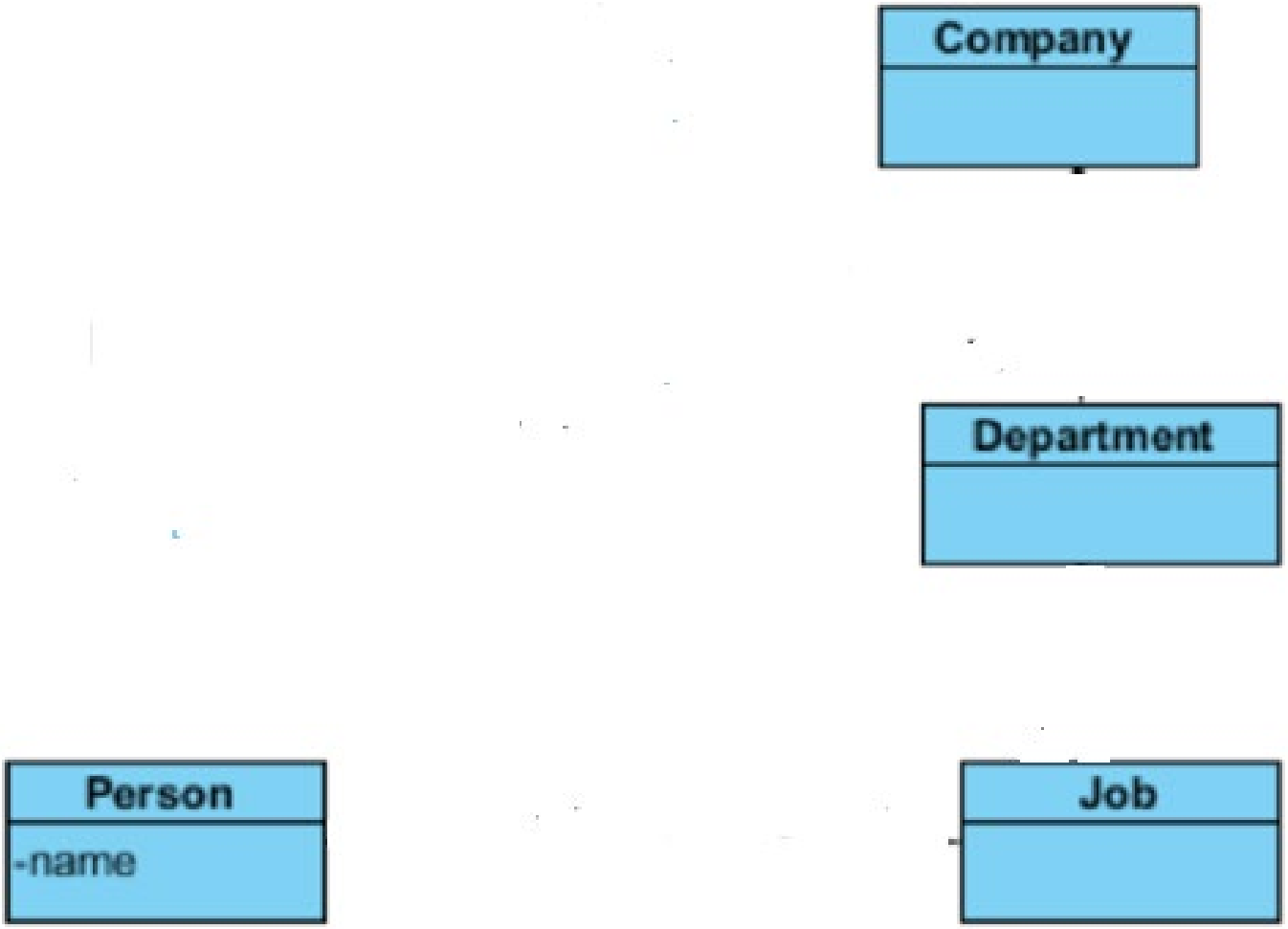


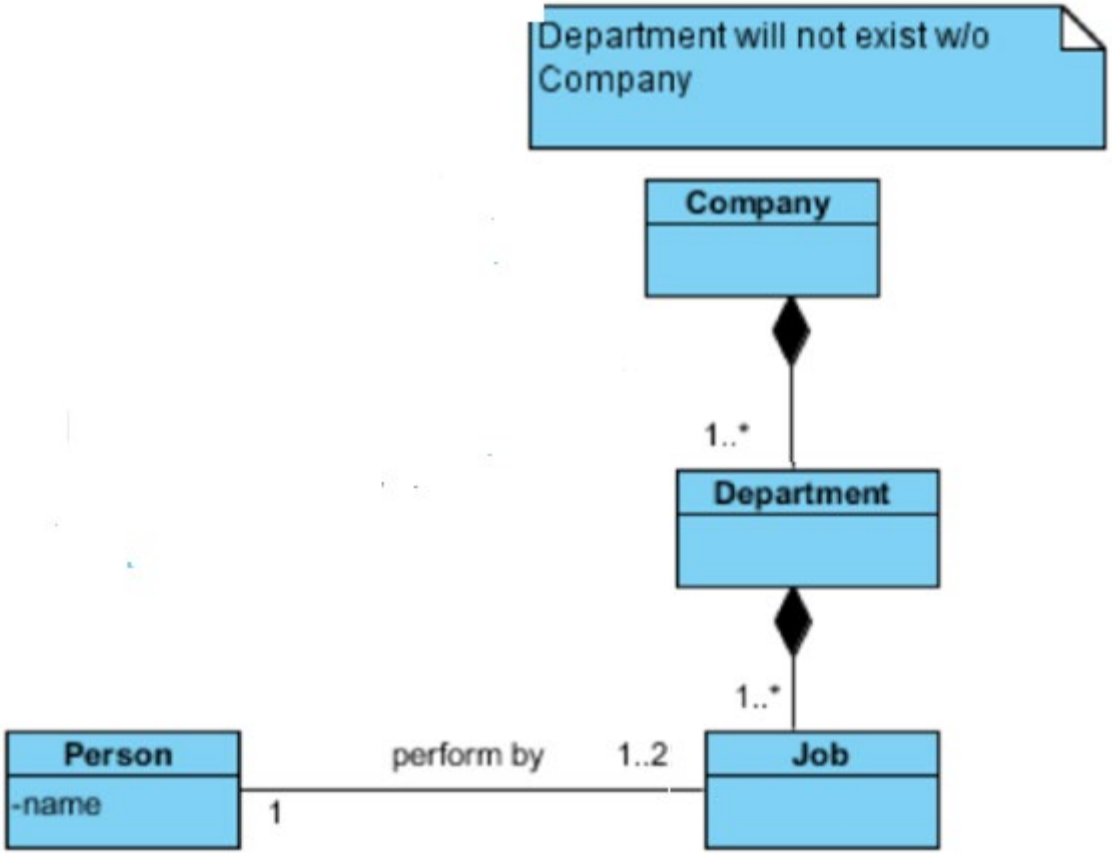




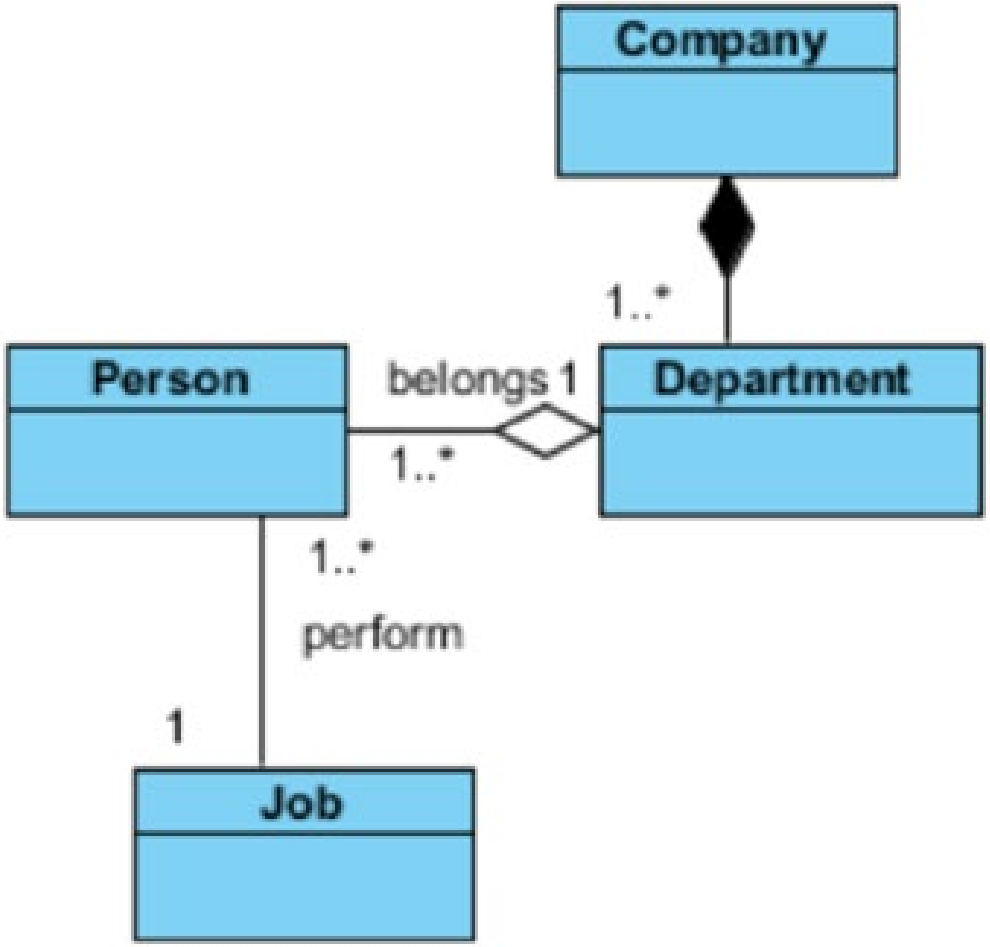


Q1Dv1: Company, Person, Department, Job



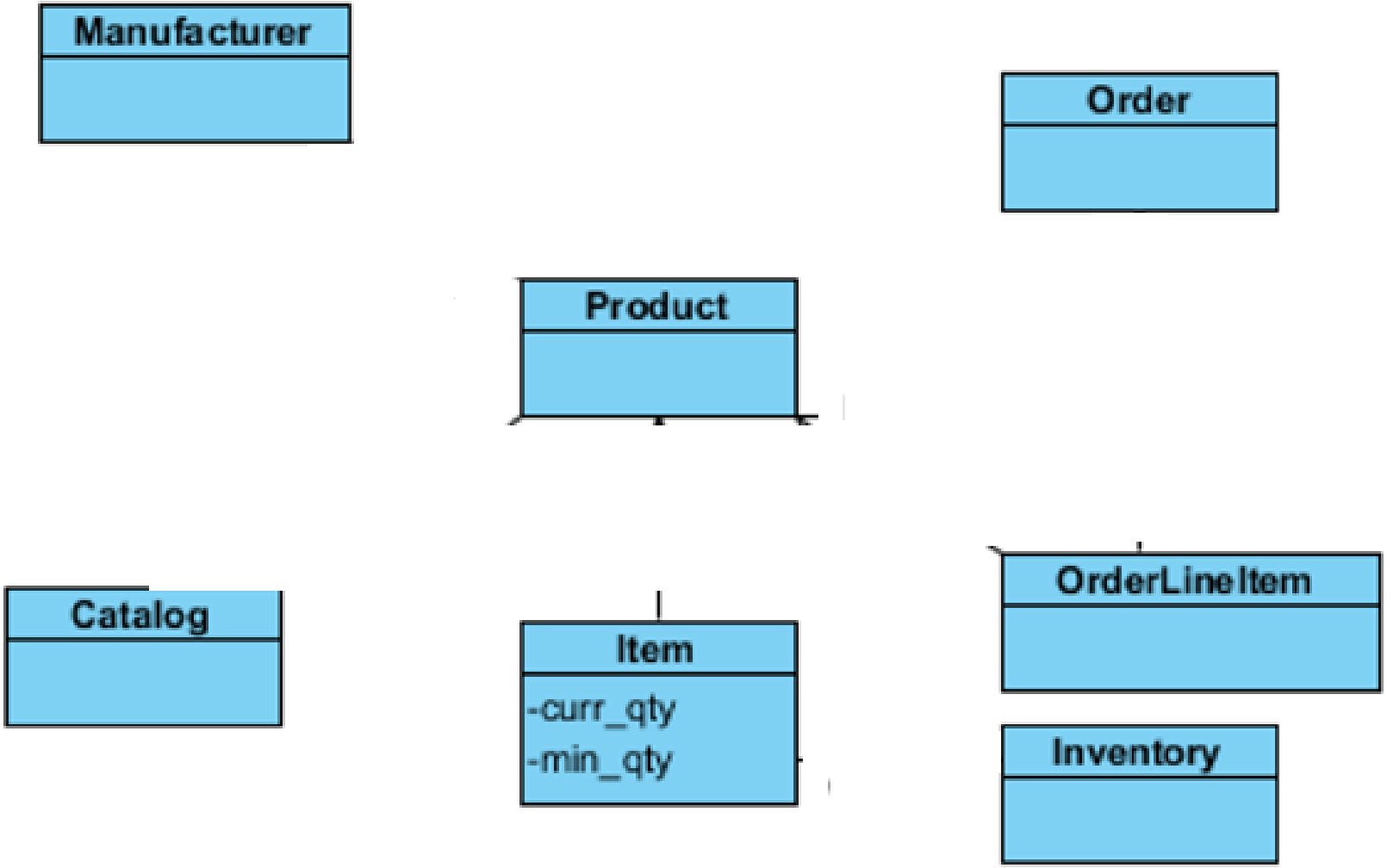

Department will not exist w/o Company
Company
1..*
Department
1..*
Person
-name
perform by
1
1..2
Job

Q1Dv2: Company, Person, Department, Job



Q1E: Product, Inventory, ItemStock, Catalog, Order, OrderLineItem, Manufacturer

(imagine an eCommerce system to browse through catalog to select the product/s to purchase. System will check whether there is still stocks in the inventory for the product you purchasing)



Inventory refers to **all the items, goods, merchandise, and materials held by a business for selling in the market** to earn a profit.

dbo.Order [Data]

Max Rows: 1000

	OrderID	CustomerID	SupplierID	StaffID	OrderDateRec...	OrderStatus	OrderPrice
▶	1	1	1	1	04/02/2019	pending	9.9900
	2	1	*NULL*	1	04/03/2019	Pending	338.7600
	3	1	1	1	04/03/2019	Completed	109.8900
	4	1	*NULL*	1	04/03/2019	Requested	9.9900
	5	1	*NULL*	1	04/03/2019	Requested	59.9000
	6	1	*NULL*	1	04/03/2019	Requested	129.7900
*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

order

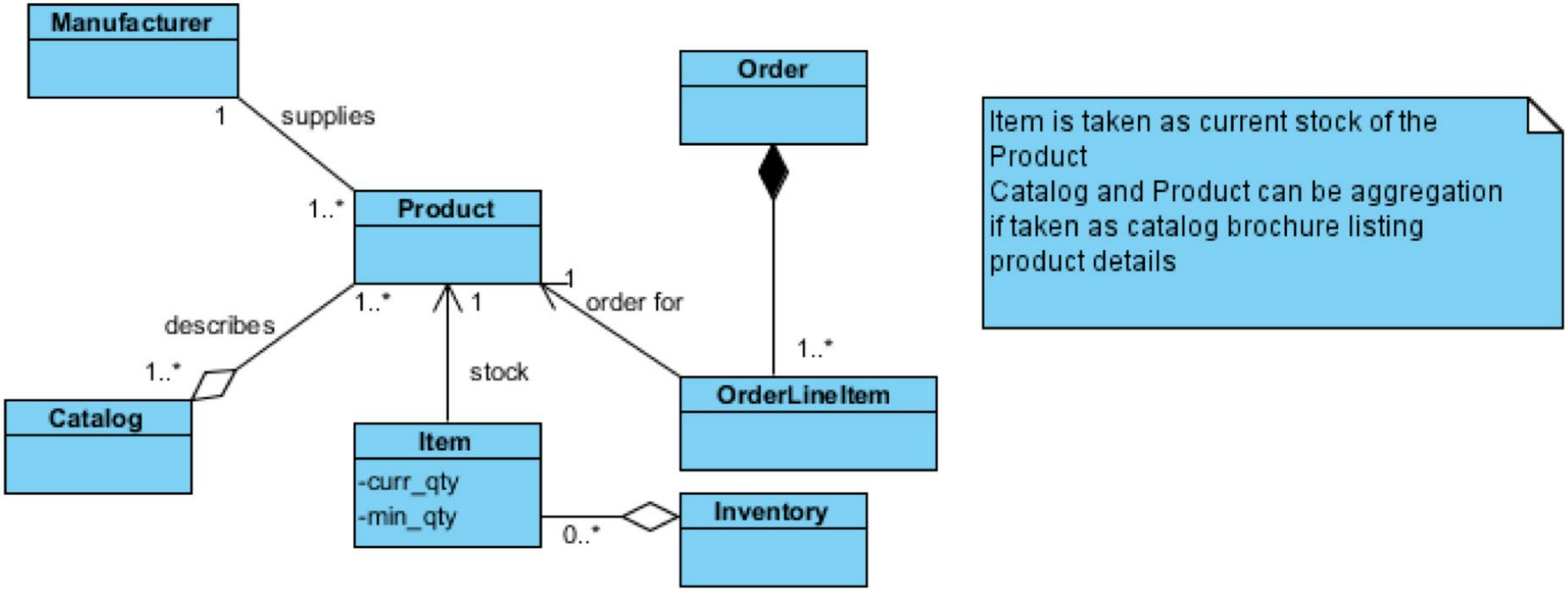
dbo.OrderLine [Data]

Max Rows: 1000

	OrderLineID	OrderID	ProductID	OrderLineQua...	OrderLinePrice
▶	1	1	1	1	9.9900
	2	2	1	1	9.9900
	3	2	1	10	99.0000
	4	2	1	11	109.8900
	5	2	1	12	119.8800
	6	3	1	1	9.9900
	7	3	1	10	99.9000
	8	4	1	1	9.9900
	9	5	2	5	9.9500
	10	5	1	5	49.9500
	11	6	1	1	9.9900
	12	6	1	10	99.9000
	13	6	2	10	19.9000
*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

OrderLine

A1. e)



- Inventory – Item stores quantity associated with Product

Q2

The following requirements describe a library system containing accounts of those users who want to access library documents. A document can be contained either directly in a library or a folder. A folder can be contained inside another folder or inside the library. Each account has its associated capability (access privilege) which provides the access levels to the different type of library items. When a user wants to access a document or a folder, his/her account's capability is checked against an access level required by the document or folder. If a user has an account, he/she can logon to the library. The user with the right capability can open, delete, and copy a folder or a document. A document can be edited also by the user. [An example of the access capability is shown Figure 1.23]

Identify the classes you will need for the library system and draw them on a UML Class Diagram. Your Class Diagram should show clearly the relationship between classes, relevant attributes (at least one) and, multiplicity, rolename, association name, if any. You may also add in the relevant methods.

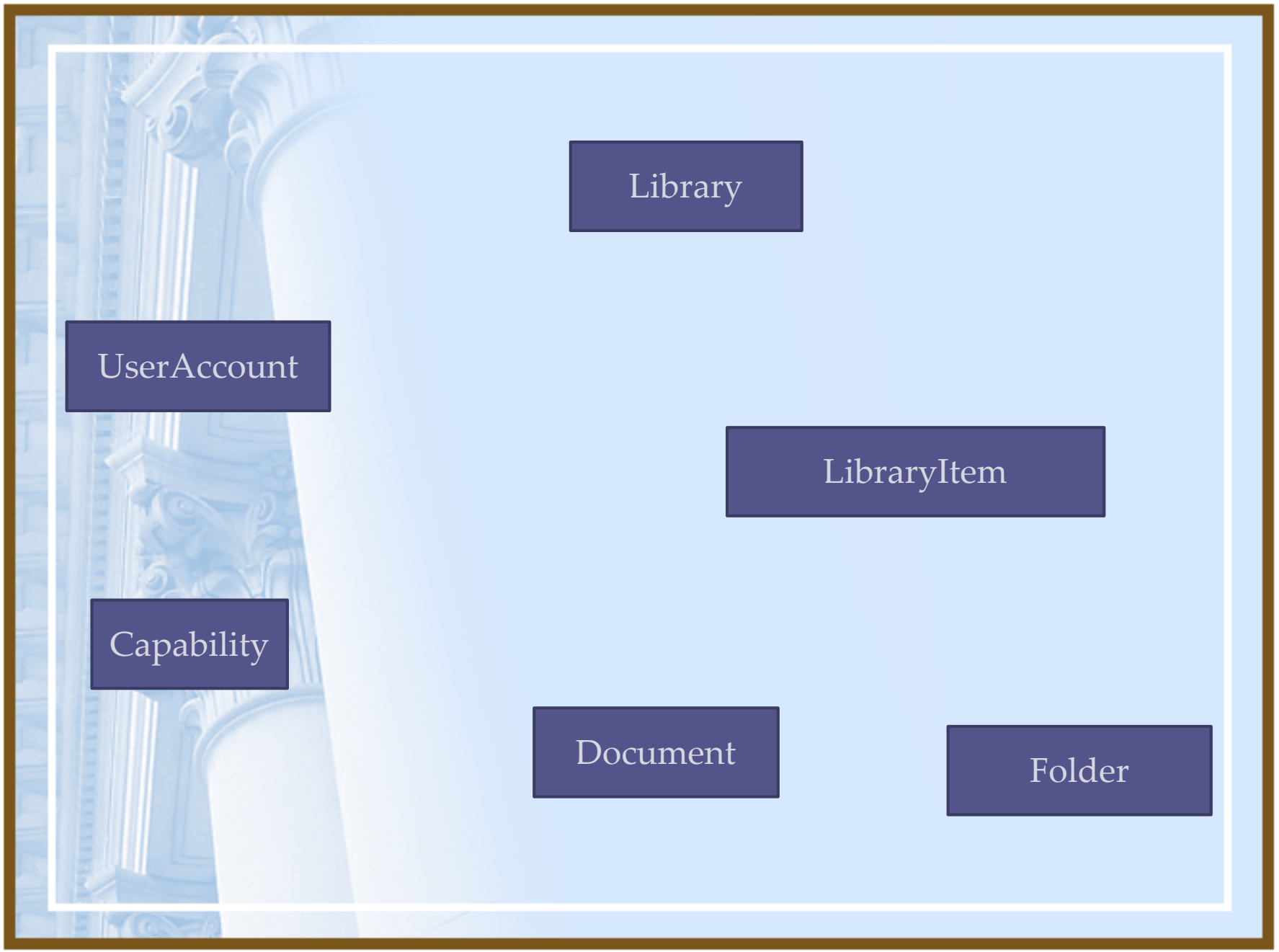
Steps:

1. Entity classes: Highlight the keywords of classes (Nouns)
   - Those Nouns which have been repeated for a few times
   - Those Nouns which have detailed description/definition
2. Relationship: Highlight the keywords of relationships (Verbs)
   - The verbs indicate “is part of” relationship
   - The verbs indicate “is a special” relationship
   - The verbs indicate normal association relationship
3. Attribute: Those nouns in description/definition of Entity classes
4. Role name: Those nouns which are instances of some entity classes in description/definition of Entity classes
5. Multiplicity: Highlight the words related to quantity
   - a
   - One to two
   - s. e.g., team members

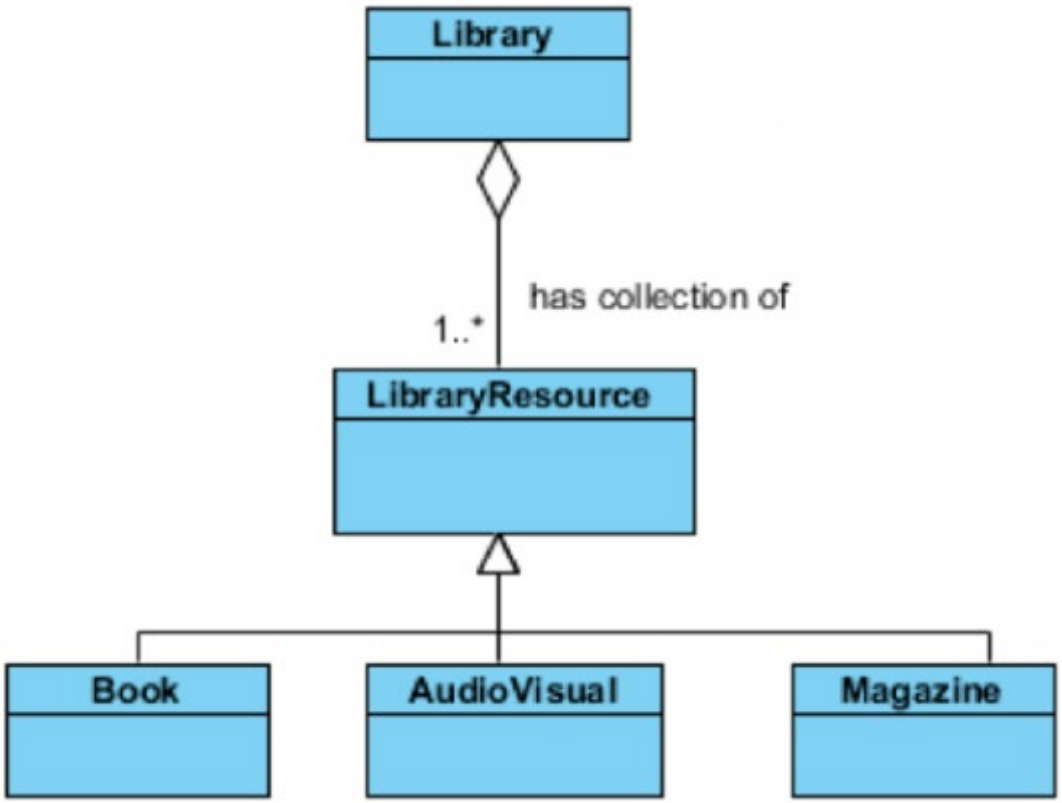
Q2

The following requirements describe a library system containing accounts of those users who want to access library documents. A document can be contained either directly in a library or a folder. A folder can be contained inside another folder or inside the library. Each account has its associated capability (access privilege) which provides the access levels to the different type of library items.

When a user wants to access a document or a folder, his/her account's capability is checked against an access level required by the document or folder. If a user has an account, he/she can logon to the library. The user with the right capability can open, delete, and copy a folder or a document. A document can be also edited by the user.


Library
UserAccount
LibraryItem
Capability
Document
Folder

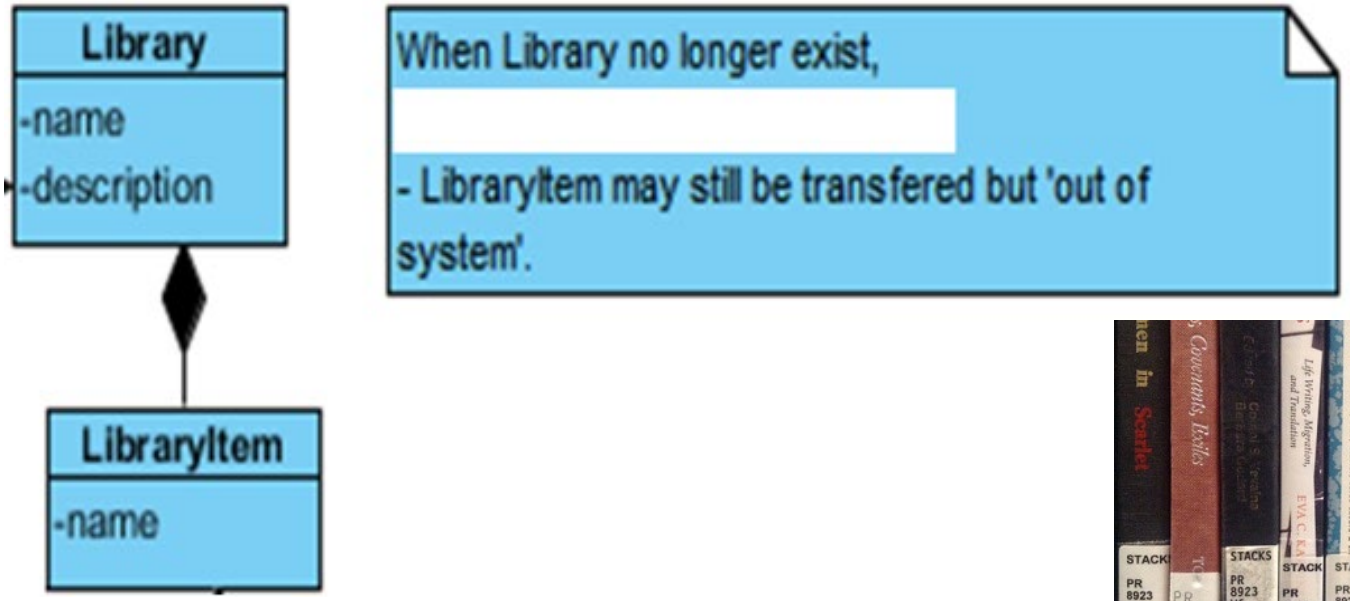
A1.a)



- Aggregation between Library and LibraryResource
- LibraryResource can be generalization or aggregation with Book, Magazine and AV.

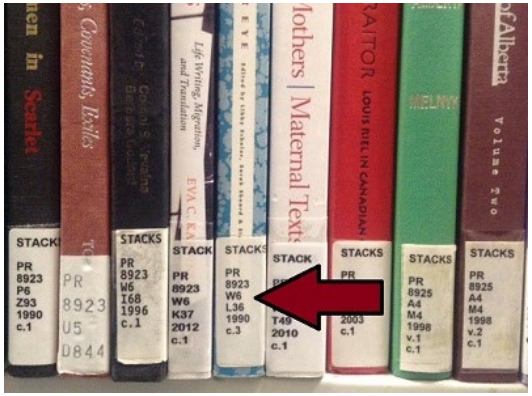
library system

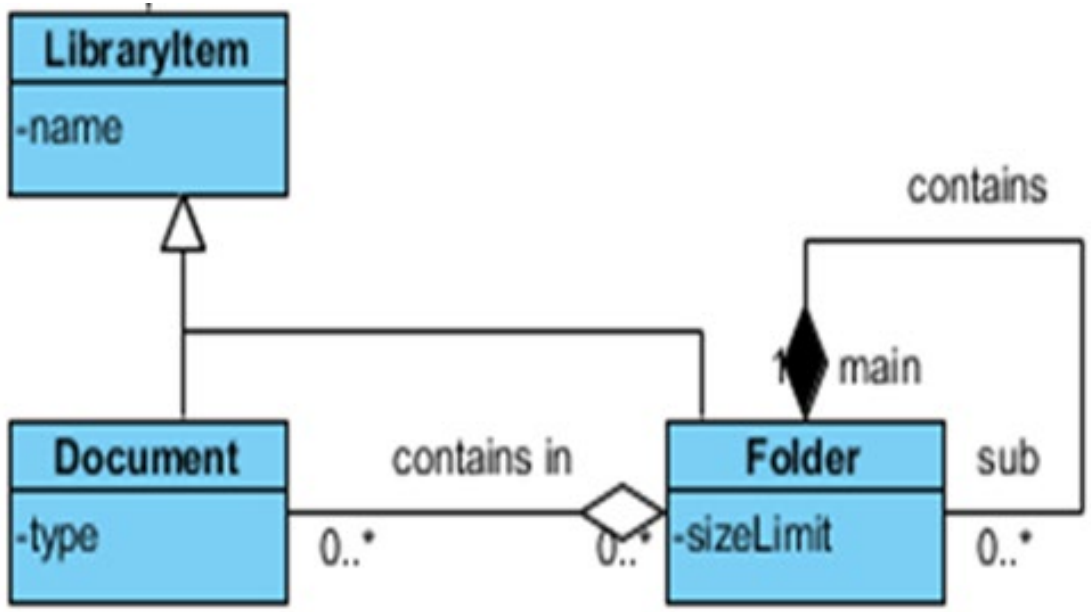
- Each account has its associated capability (access privilege) which provides the access levels to the different type of library items.



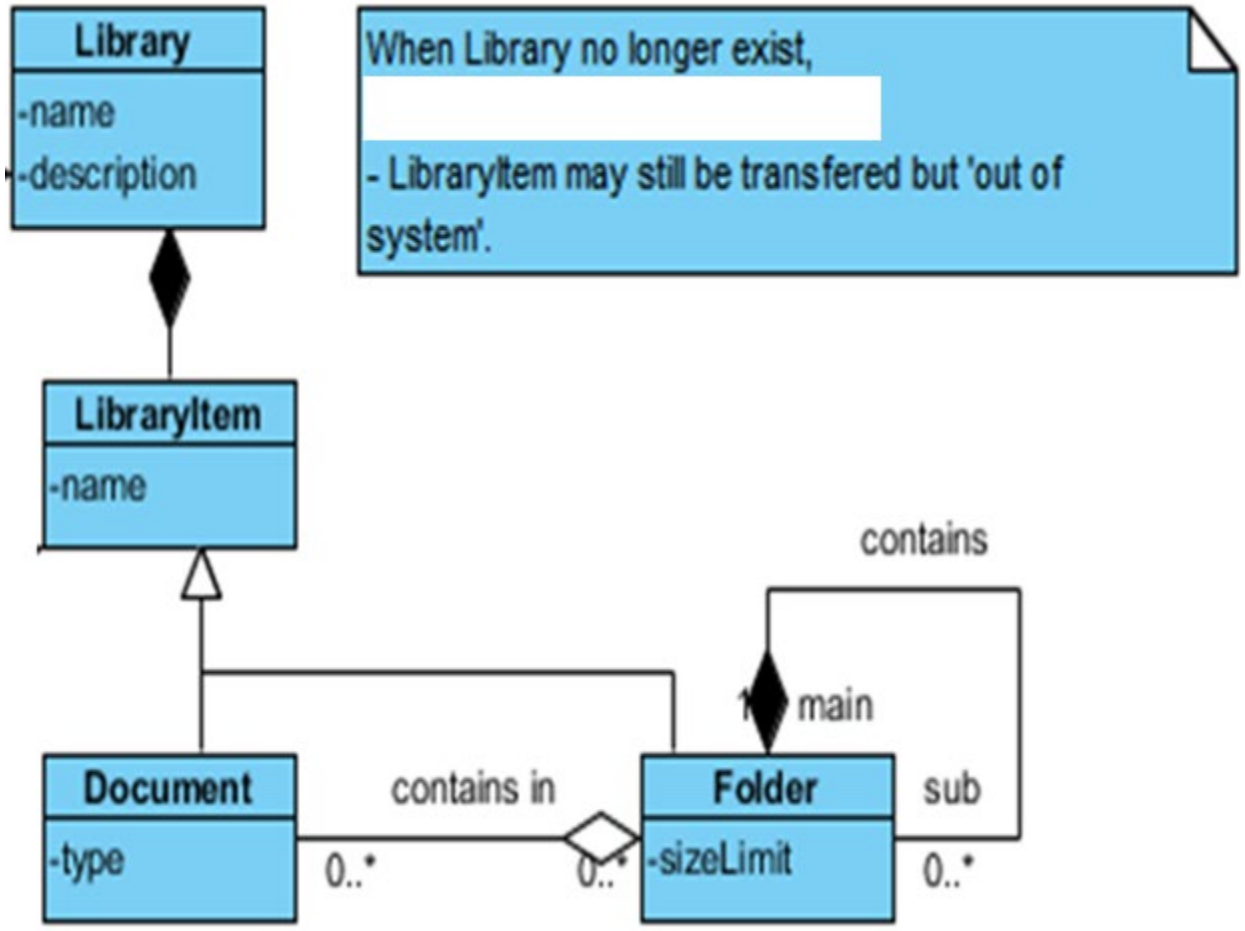
Items managed in the library system

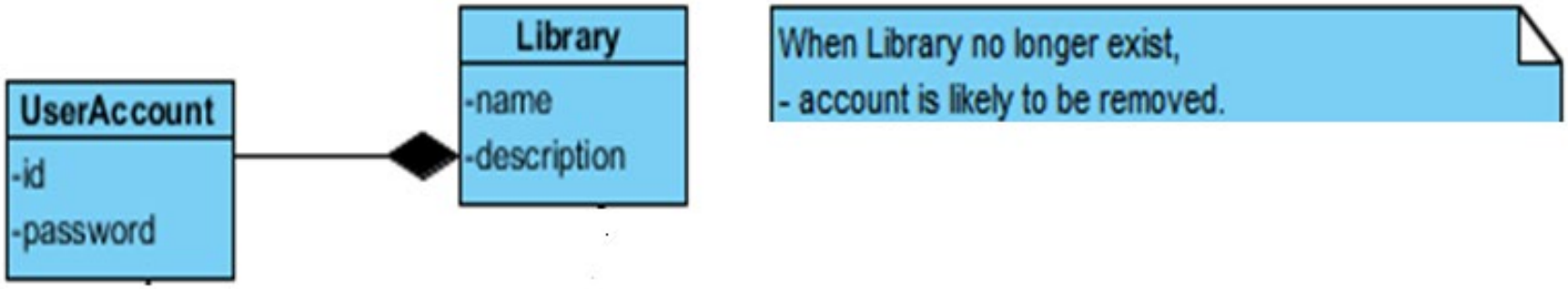
Call number



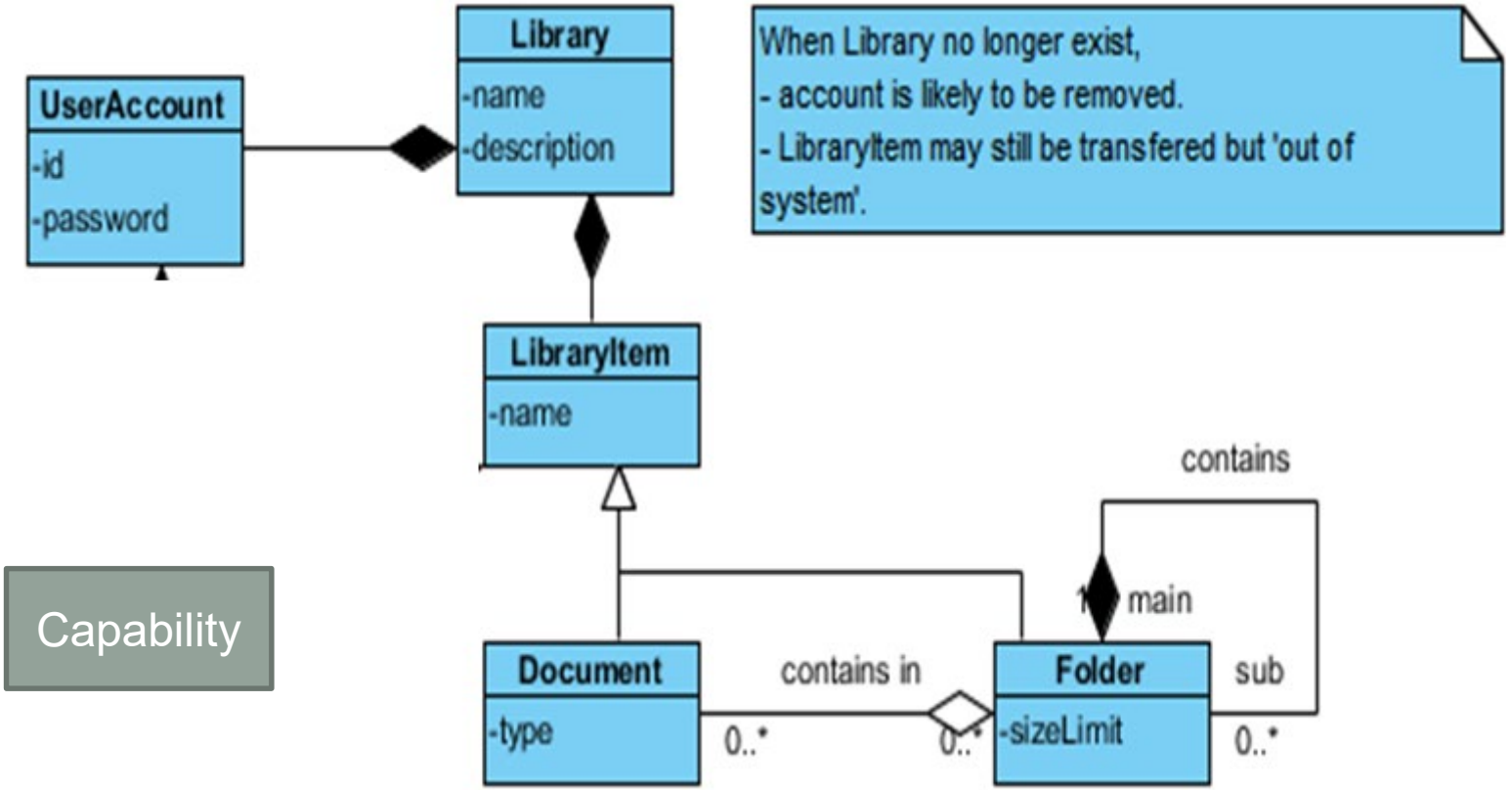


A document can be contained either directly in a library or a folder.
A folder can be contained inside another folder or inside the library.


Library
-name
-description
When Library no longer exist,
- LibraryItem may still be transfered but 'out of system'.
LibraryItem
-name
contains
main
Document
-type
contains in
0..*
0..*
Folder
-sizeLimit
sub
0..*



a library system containing accounts of those users



Each account has its associated capability (access privilege) which provides the access levels to the different type of library items.

Access Capabilities	
No Access	No access permission granted
Read (R)	Read but make no changes
Write (W)	Write to file. Includes change capability
Execute (X)	Execute a program
Delete (D)	Delete a file
Change (C)	Read, write, execute, and delete. May not change file permission.
List (L)	List the files in a directory
Full Control (FC)	All abilities. Includes changing access control permissions.

Access Permissions	
Public	R – L
Group	R – X
Owner	R – W – X – D
Admins	FC
System	FC

Figure 1.23 An example of access permissions. Access permissions are applied to an object based on the level of clearance given to a subject.

ACCESS CONTROL LIST

Mary:
UserMary Directory - FullControl
UserBob Directory - Write
UserBruce Directory - Write
Printer 001 – Execute

Bob:
UserMary Directory - Read
UserBob Directory - Full Control
UserBruce Directory - Write
Printer 001 – Execute

Bruce:
UserMary Directory - No Access
User Bob Directory - Write
UserBruce Directory - Full Control
Printer 001 – Execute

Sally:
UserMary Directory - No Access
UserBob Directory - No Access
UserBruce Directory - No Access
Printer 001 - No Access

ACCESS CONTROL LIST
(Continued)

Group Administrators:
- Members- Ted, Alice
- UserBruce Directory - Full Control
- UserSally Directory - Full Control
- UserBob Directory - Full Control
- UserMary Directory - Full Control

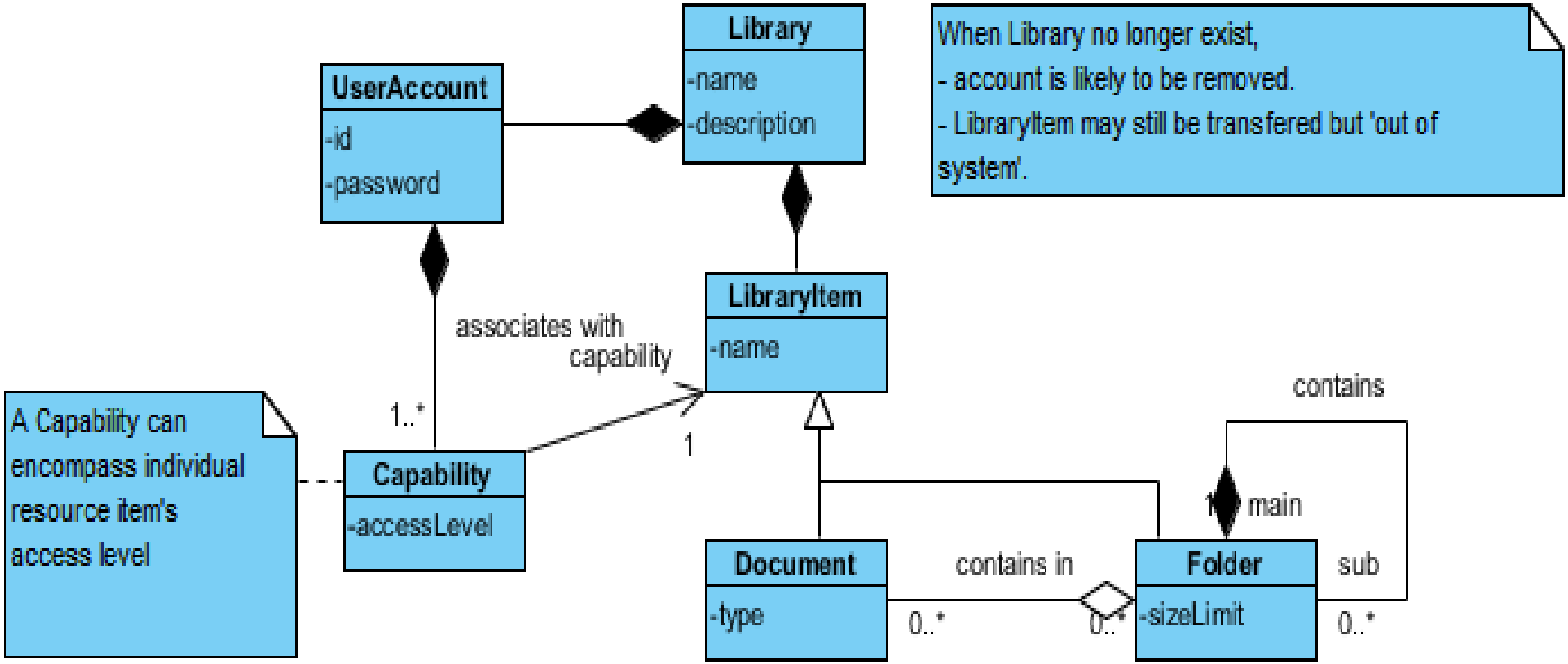
Group Printer Users:
- Members – Bruce, Sally, Bob
- UserBruce Directory – No Access
- UserSally Directory - No Access
- UserBob Directory - No Access
- PrinterDevice P1 – Print
- PrinterDevice P2 – Print
- PrinterDevice P3 – Print

- Example Capability List (Access Matrix)

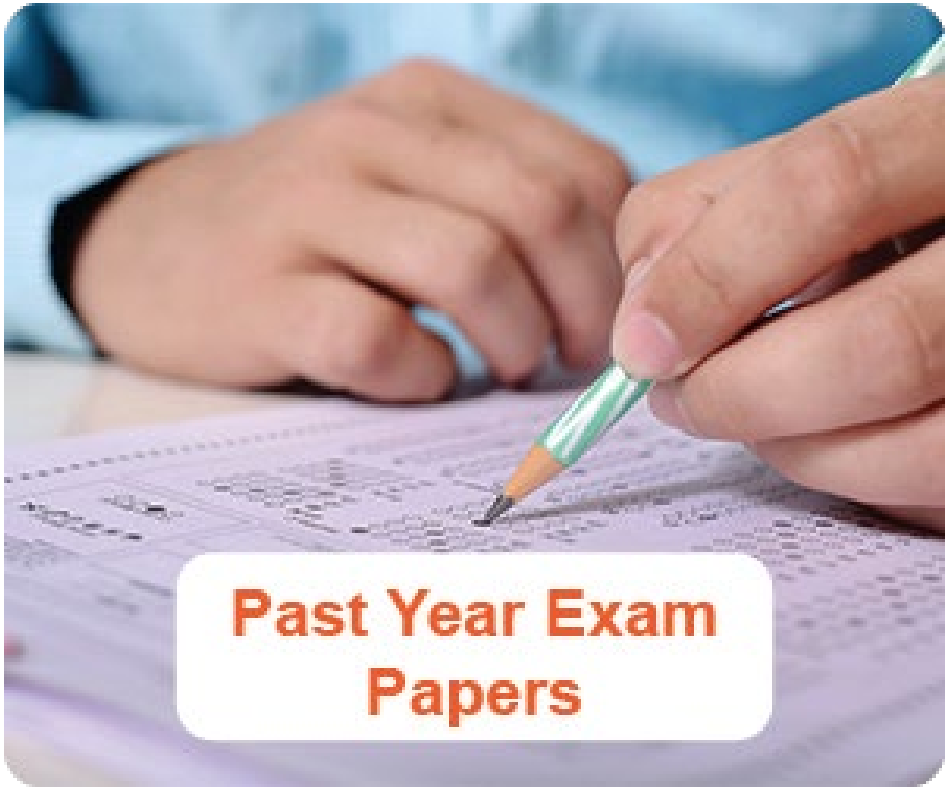
	Object		
Subject	**File1**	**File2**	**Folder1**
Alice	**rw**	**r**	**rw**
Bob	**r**	**r**	**r**
Cindy	**rw**	**rw**	**r**
Dan	..	..	..

- Capability list from the perspective of subject.
- A User can have multiple *capabilities*. Each *capability* defines its access level to different resources (document/folder).

  Eg, Alice 's capabilities (File1, rw), (File2, r), (Folder1, rw)


Library
-name
-description
When Library no longer exist,
- account is likely to be removed.
- LibraryItem may still be transfered but 'out of system'.
UserAccount
-id
-password
associates with capability
LibraryItem
-name
1..*
1
A Capability can encompass individual resource item's access level
Capability
-accessLevel
contains
1
main
Document
-type
contains in
Folder
-sizeLimit
sub
0..*
0..*
0..*

PYP revision



2017-2018S1Q4a

Past Year Paper 2017-2018S1 Q4a

- (a) Consider a car racing event application, CREA, with the following requirements:

- The car racing event is held every year. Each year's event consists of one
- or more races. A race has the information of the location, the date, the road length and also the weather on the race date. The location will record the country, city and venue where the race is held.

- One or more teams participate in each year's event. Each team has a team ID, the team name and the founding date of the team. Each team has one or more drivers participating in the each year's event. Each team has one or more members. The members include one manager, one or more drivers and one or more supporting staff. The member ID, the name, and the date of birth of member are recorded. Additionally, the driver's driving experience and past achievements are also recorded.

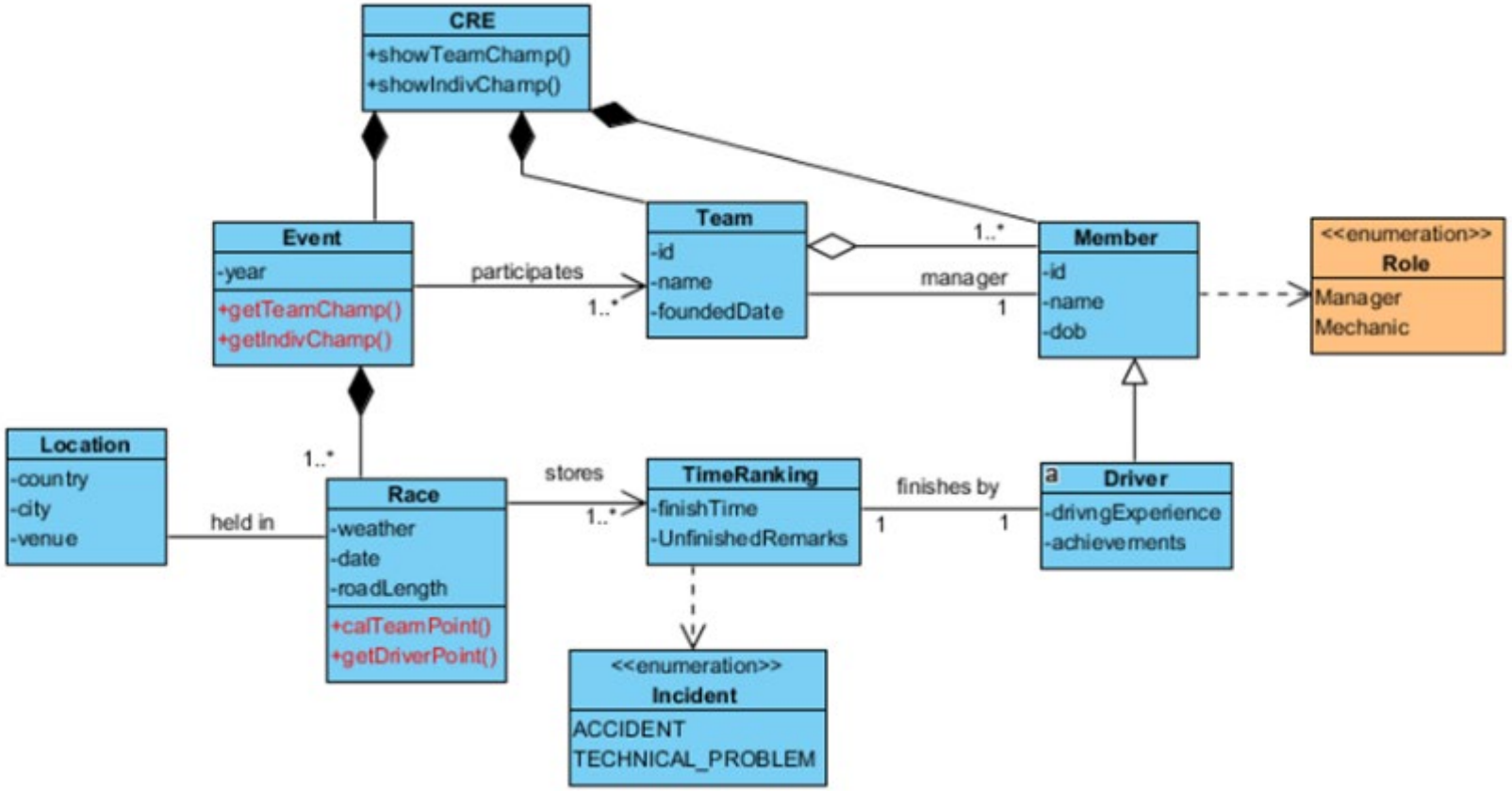
- For each race, the time taken by each driver to finish the race is recorded. If the driver fails to finish a race, the cause of the unfinished race is recorded. The cause can be due to accidents or technical problems of his/her car. Every driver who finishes a race will be given a certain number of points based on his/her ranking (in terms of finish time) in the race.

- After a year's races are finished, the team with the highest total points earned by its drivers during the year would be given the team championship of the year. The driver who earns the highest total points will be given an individual championship.

- You are tasked to identify the entity classes needed to build the application based on the description above.

- Show your design in a Class Diagram. Your Class Diagram should show clearly the relationship between classes, relevant attributes (at least TWO), logical multiplicities, meaningful role names, association names and constraint, if any. For methods, you only need to show the method(s) which can be used to derive the individual and team championships.

- (15 marks)



Correct classes : **5 marks** (esp Race, Driver, TimeRanking, Team, Location, Member, Event)
Appropriate methods in Event, race class : 1 mark
Incident/Role as enum : 1 mark
Correct use of association, composition : **2 marks**
Inheritance for Driver : 1 mark
Attributes **: 2 marks** (no private -1)
Correct logicl Multiplicity : **2 marks** (0..*/1..*)
Association name : 1 mark