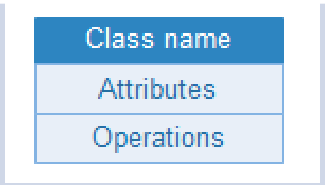
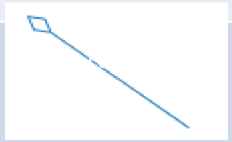
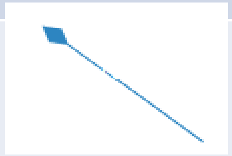
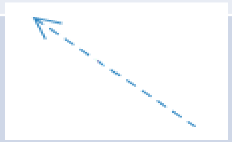
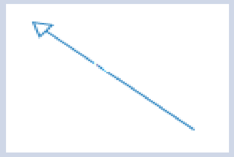
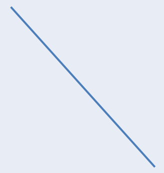
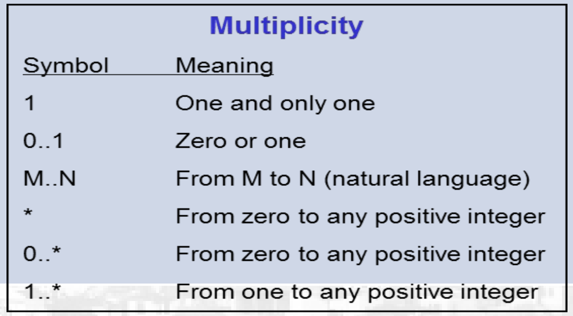
Class Diagram

- A class diagram depicts classes and their interrelationships
- Used for describing structure and behavior in the use cases
- Provide a conceptual model of the system in terms of entities and their relationships
- Used for requirement capture, end-user interaction
- Detailed class diagrams are useful for software developers

Class Diagram

S.No	Name	Description	Notation
1	Classes and interface	They are used to show the different objects in a system, their attributes, their operations and the relationships among them.	
2	Object	An object is an instance or occurrence of a class.	Object: Class
3	Aggregation	An aggregation describes a group of objects and how you interact with them.	
4	Composition	Composition represents whole-part relationships and is a form of aggregation.	
5	Dependency	Dependency relationship is a relationship in which one element, the client, uses or depends on another element, the supplier.	

Class Diagram

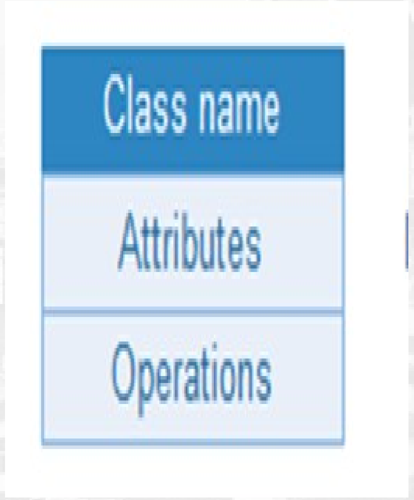
S.No	Name	Description	Notation
3	Generalization	Generalization is a relationship in which one model element (the child) is based on another model element (the parent).	
4	Association	Association is a relationship between two classifiers, such as classes or use cases, that describes the reasons for the relationship and the rules that govern the relationship.	
5	Multiplicity		

Multiplicity

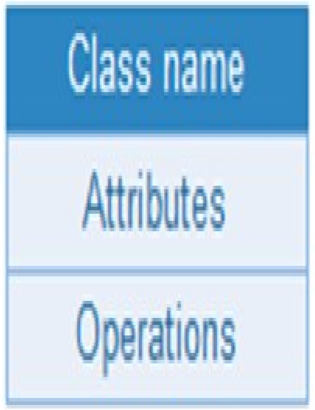
Symbol	Meaning
1	One and only one
0..1	Zero or one
M..N	From M to N (natural language)
*	From zero to any positive integer
0..*	From zero to any positive integer
1..*	From one to any positive integer

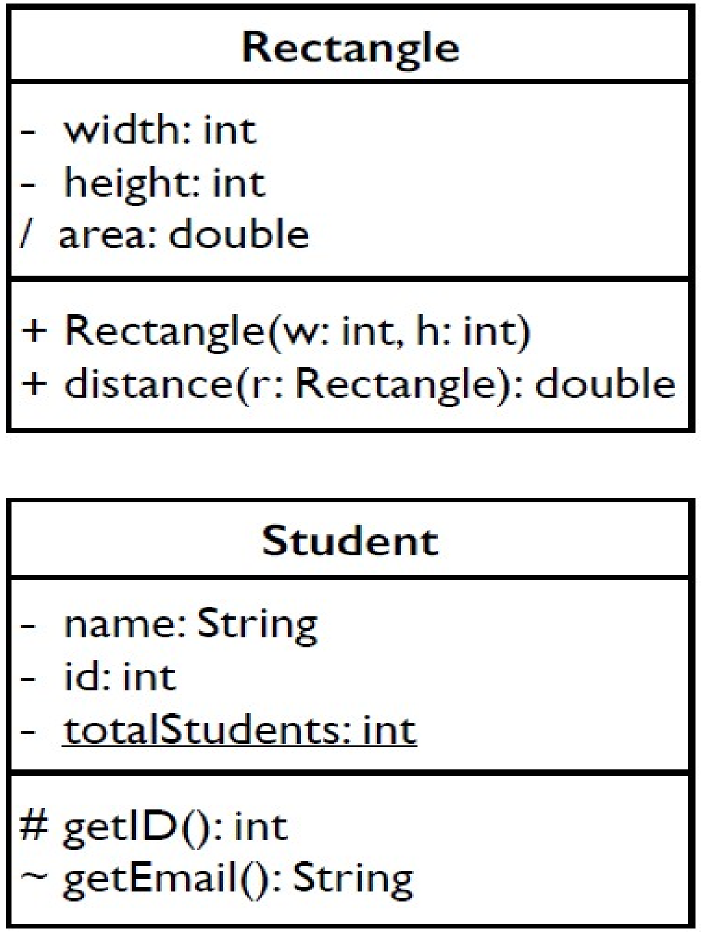
Drawing a Class Diagram ?

- Identify and model classes—Which classes do we need?
- Identify and model associations—How are the classes connected?
- Define attributes—What do we want to know about the objects?

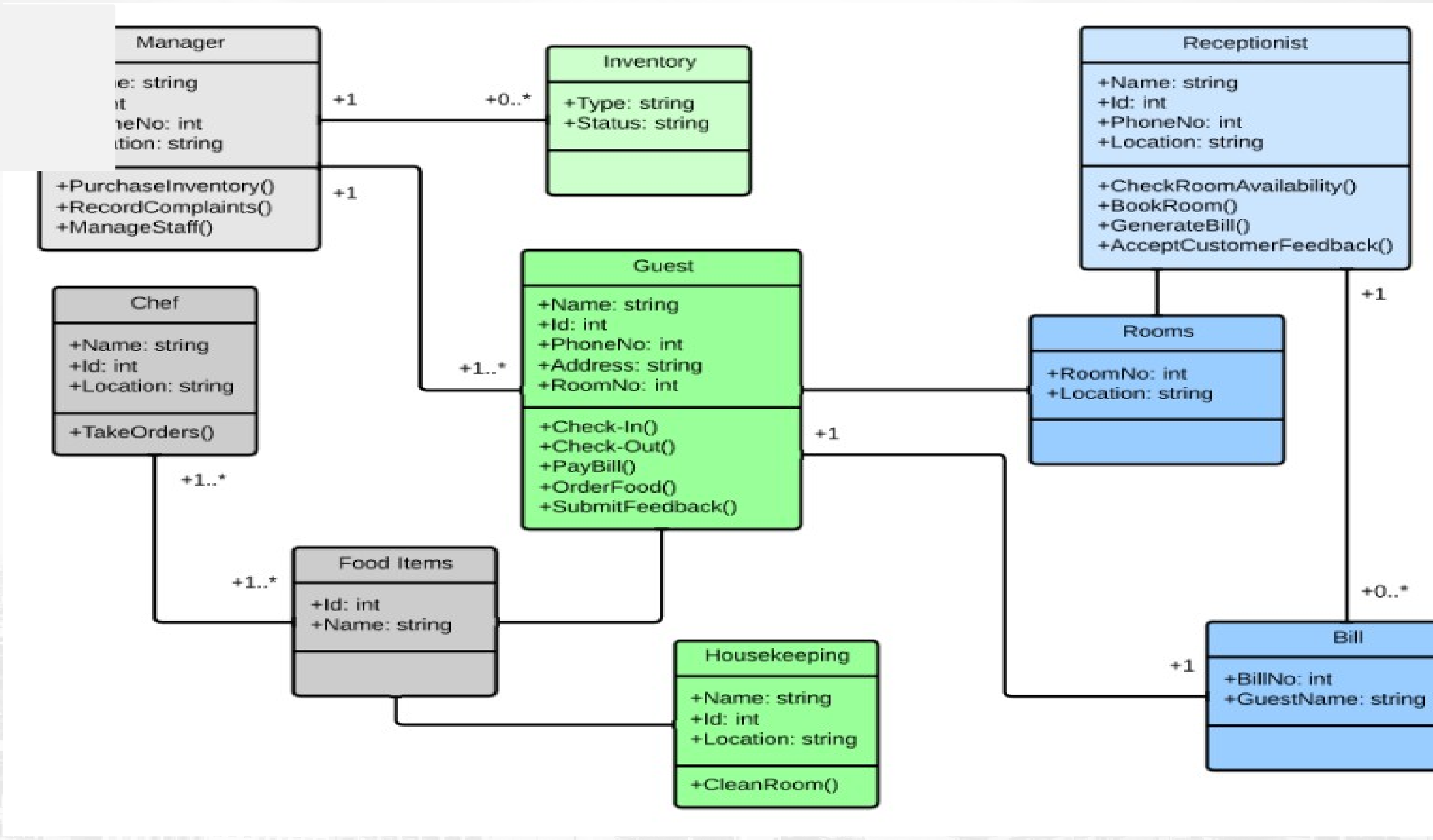


A Single Class

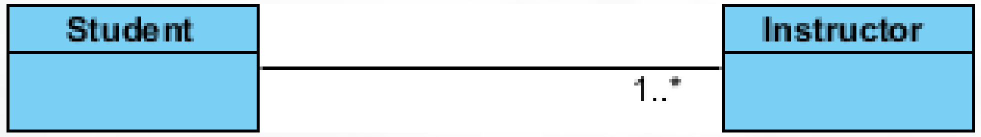




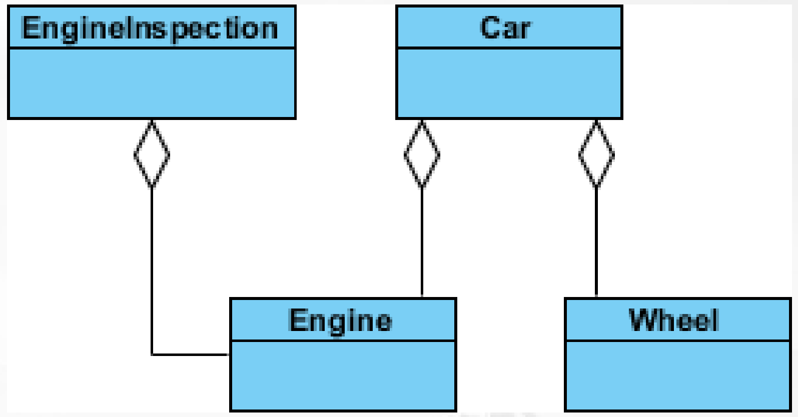
Notation	Visibility Name
-	Private
+	Public
#	Protected
~	Package/default


Manager
e: string
t
neNo: int
tion: string
+PurchaseInventory()
+RecordComplaints()
+ManageStaff()
+1
+0..*
Inventory
+Type: string
+Status: string
+1
+1..*
Receptionist
+Name: string
+Id: int
+PhoneNo: int
+Location: string
+CheckRoomAvailability()
+BookRoom()
+GenerateBill()
+AcceptCustomerFeedback()
+1
Chef
+Name: string
+Id: int
+Location: string
+TakeOrders()
Guest
+Name: string
+Id: int
+PhoneNo: int
+Address: string
+RoomNo: int
+Check-In()
+Check-Out()
+PayBill()
+OrderFood()
+SubmitFeedback()
Rooms
+RoomNo: int
+Location: string
+1
+1..*
+1..*
Food Items
+Id: int
+Name: string
+0..*
Bill
+BillNo: int
+GuestName: string
+1
Housekeeping
+Name: string
+Id: int
+Location: string
+CleanRoom()

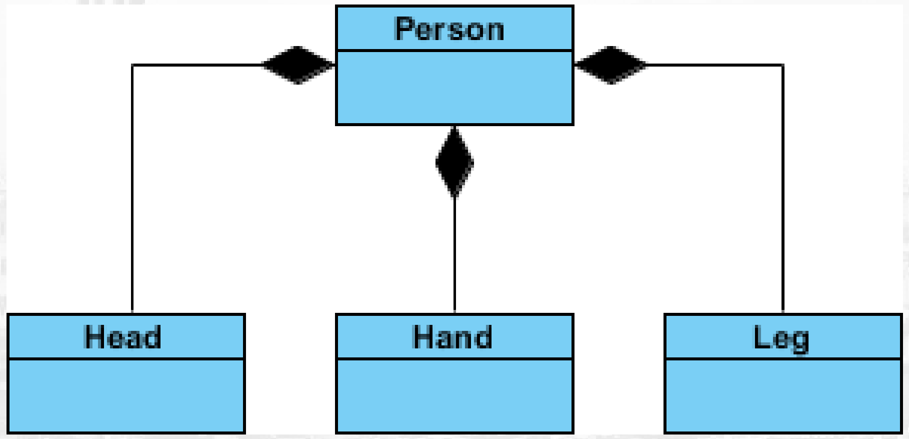
Types of Relations in Class Diagram



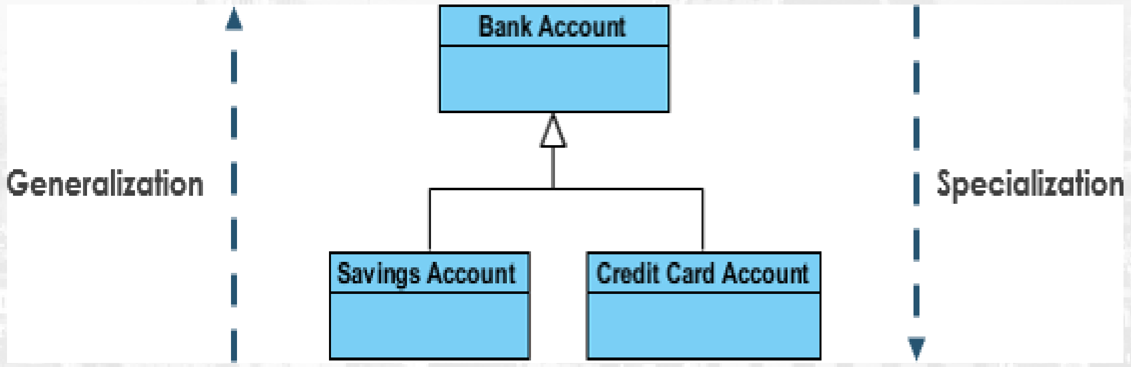
Association



Aggregation (exist alone)

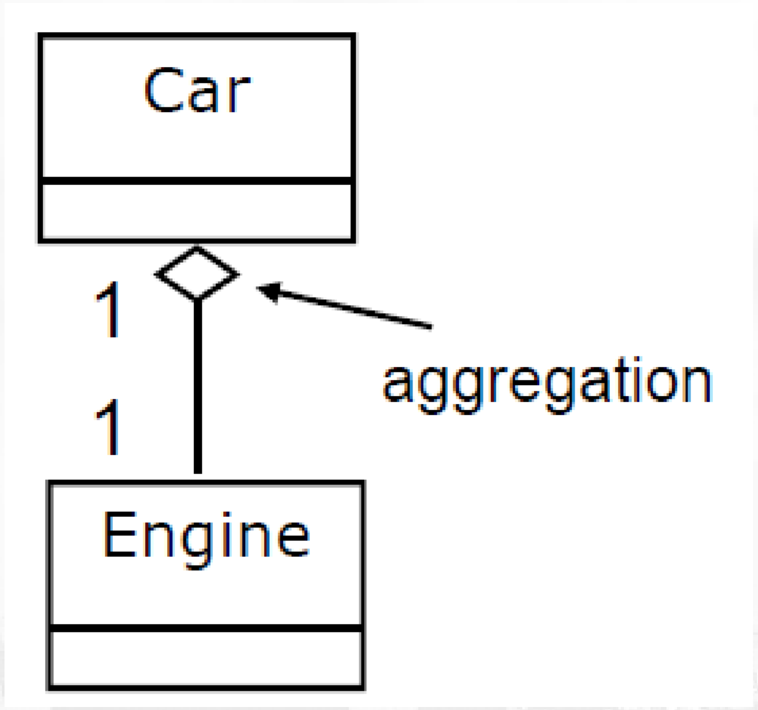


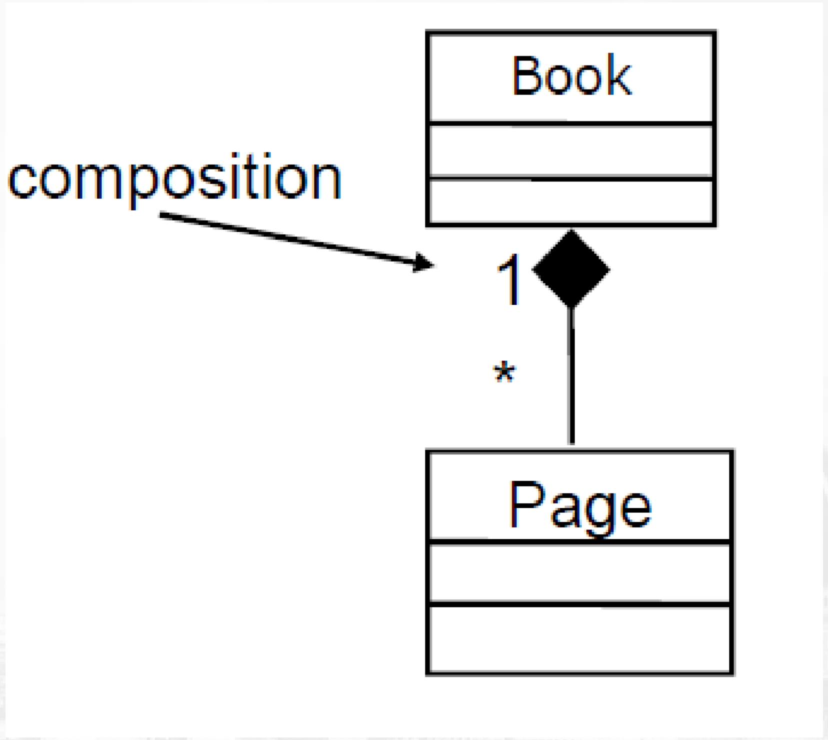
Composition

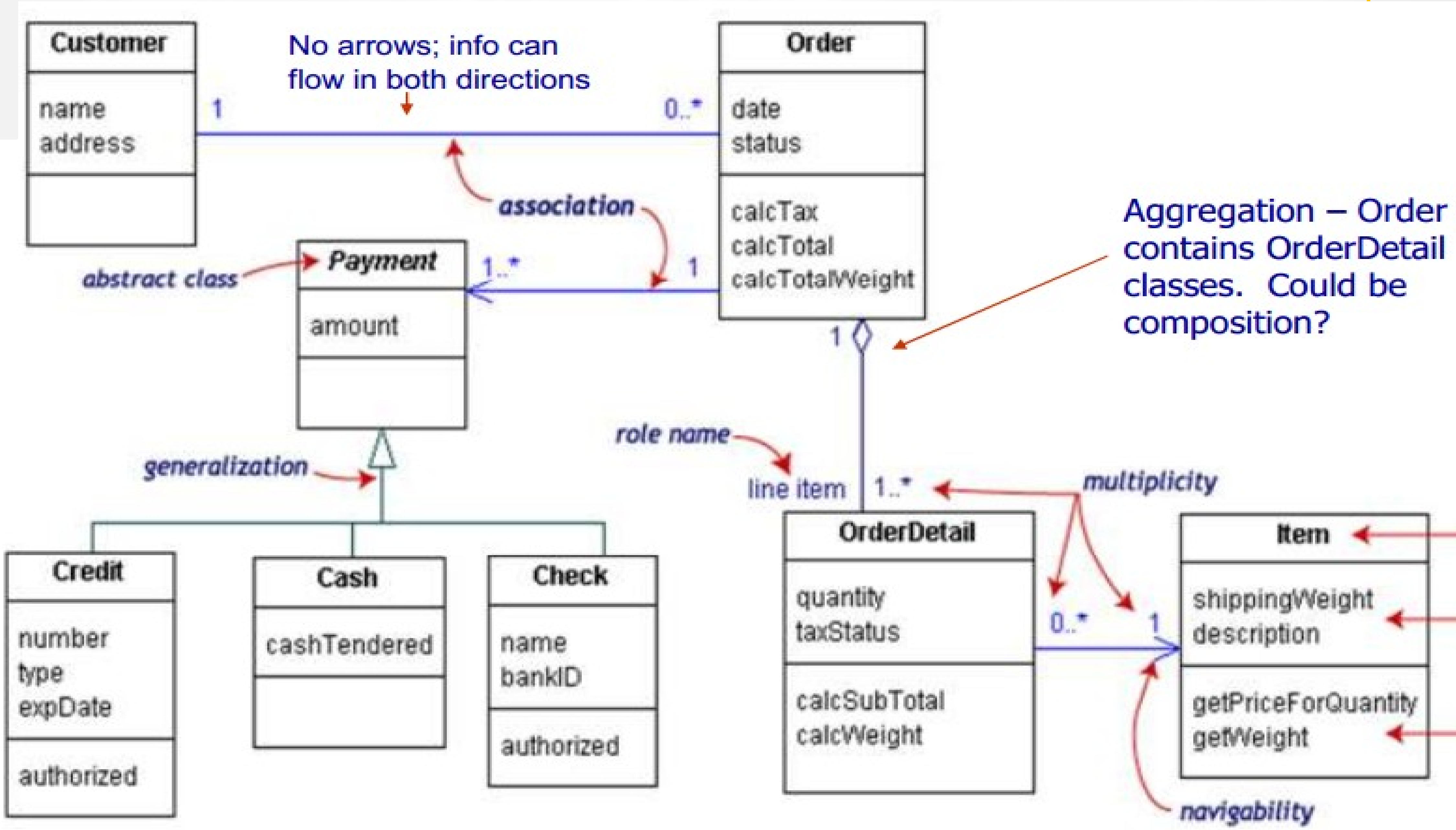


Generalization

Another Example



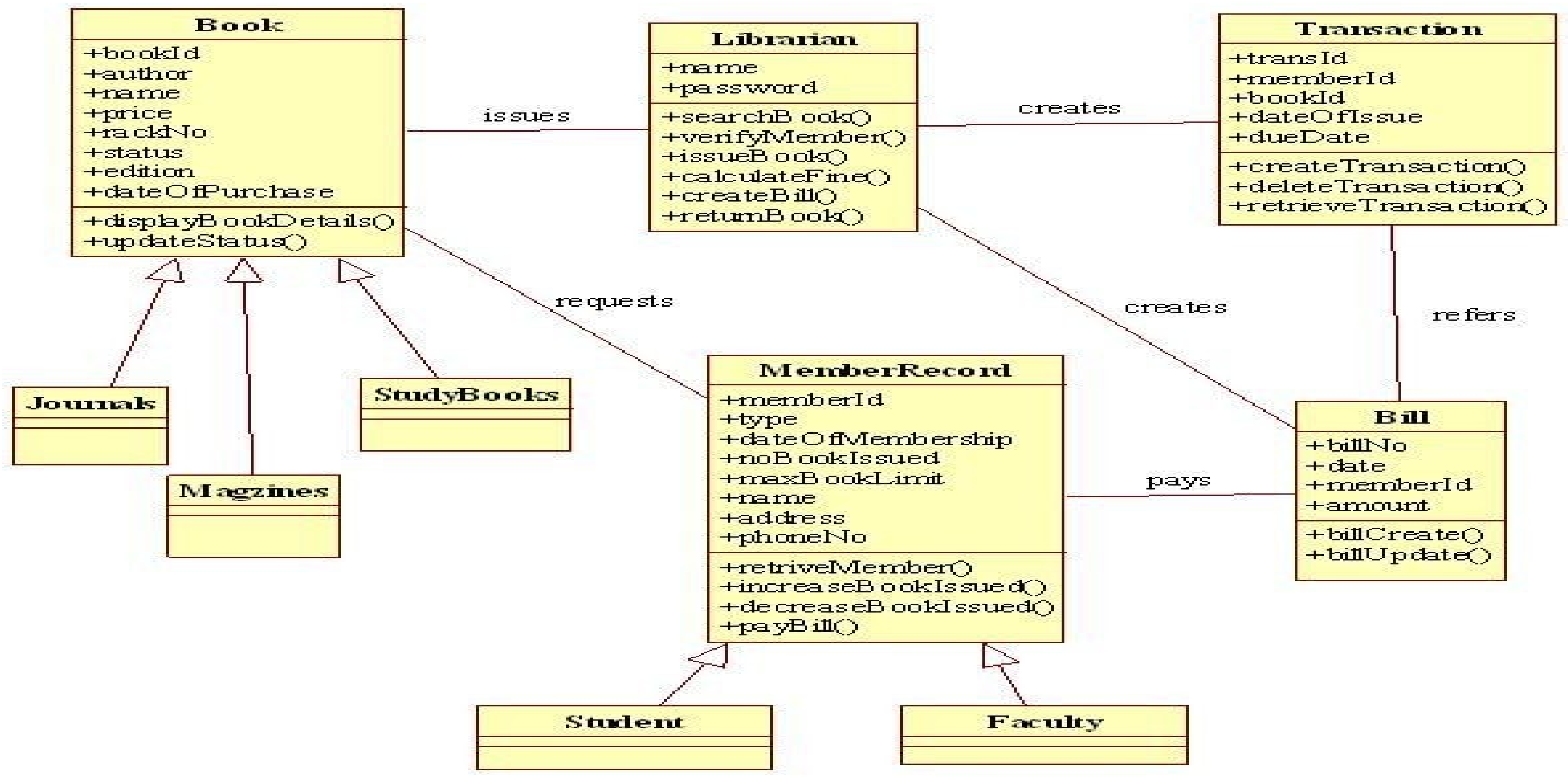




No arrows; info can flow in both directions

Aggregation – Order contains OrderDetail classes. Could be composition?

Example : Library Management System



Benefits of class diagrams

- To understand the general overview of plan of an application.
- Illustrate data models for information systems, no matter how simple or complex.
- Visually express any specific needs of a system.
- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Construction of software applications using object oriented languages.

References

- Ian Sommerville, ― Software Engineering‖, Addison and Wesley. 9th Ed., 2011.
- Roger S Pressman, Software Engineering: A Practitioner's Approach, Mcgraw-Hill, ISBN: 0073375977, Seventh Edition, 2014
- Pankaj Jalote, Software Engineering: A Precise Approach, Wiley India.2010.

Thank You