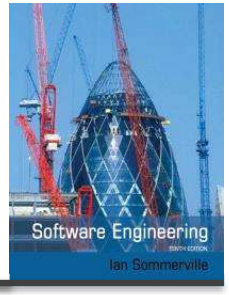
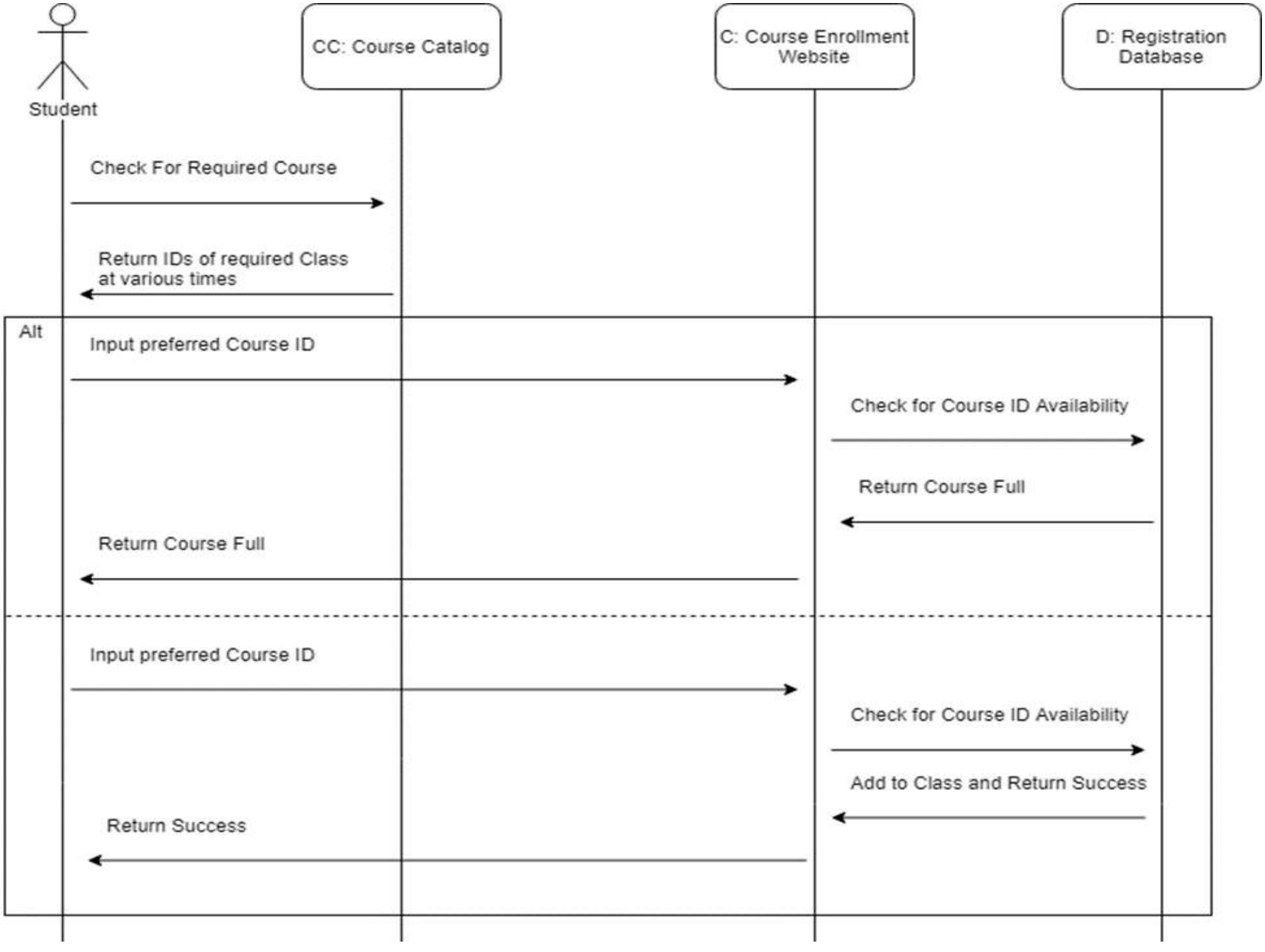
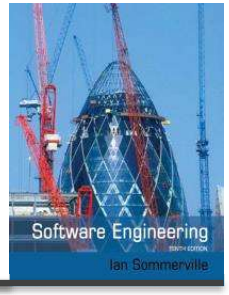
Sequence Diagram



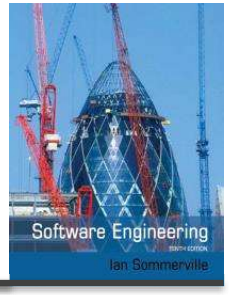
- a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrollment, so the registration process must include checks that places are available. Assume that the student accesses an electronic course catalog to find out about available courses.


Student
CC: Course Catalog
C: Course Enrollment Website
D: Registration Database
Check For Required Course
Return IDs of required Class at various times
Alt
Input preferred Course ID
Check for Course ID Availability
Return Course Full
Return Course Full
Input preferred Course ID
Check for Course ID Availability
Add to Class and Return Success
Return Success



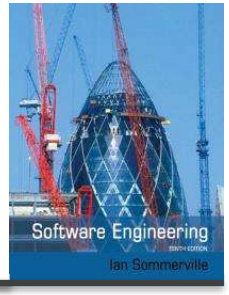
Structural models

Structural models



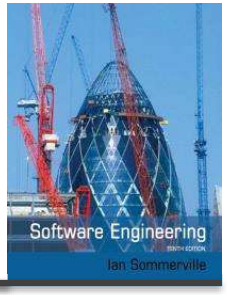
- Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.
- Structural models may be static models, which show the structure of the system design, or dynamic models, which show the organization of the system when it is executing.
- You create structural models of a system when you are discussing and designing the system architecture.

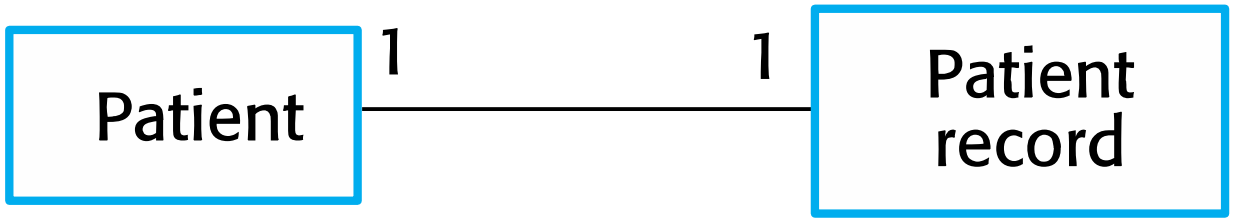
Class diagrams



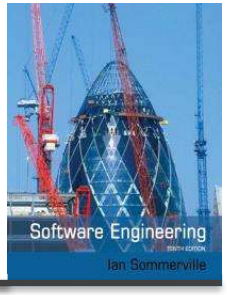
- Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- An object class can be thought of as a general definition of one kind of system object.
- An association is a link between classes that indicates that there is some relationship between these classes.
- When you are developing models during the early stages of the software engineering process, objects represent something in the real world, such as a patient, a prescription, doctor, etc.

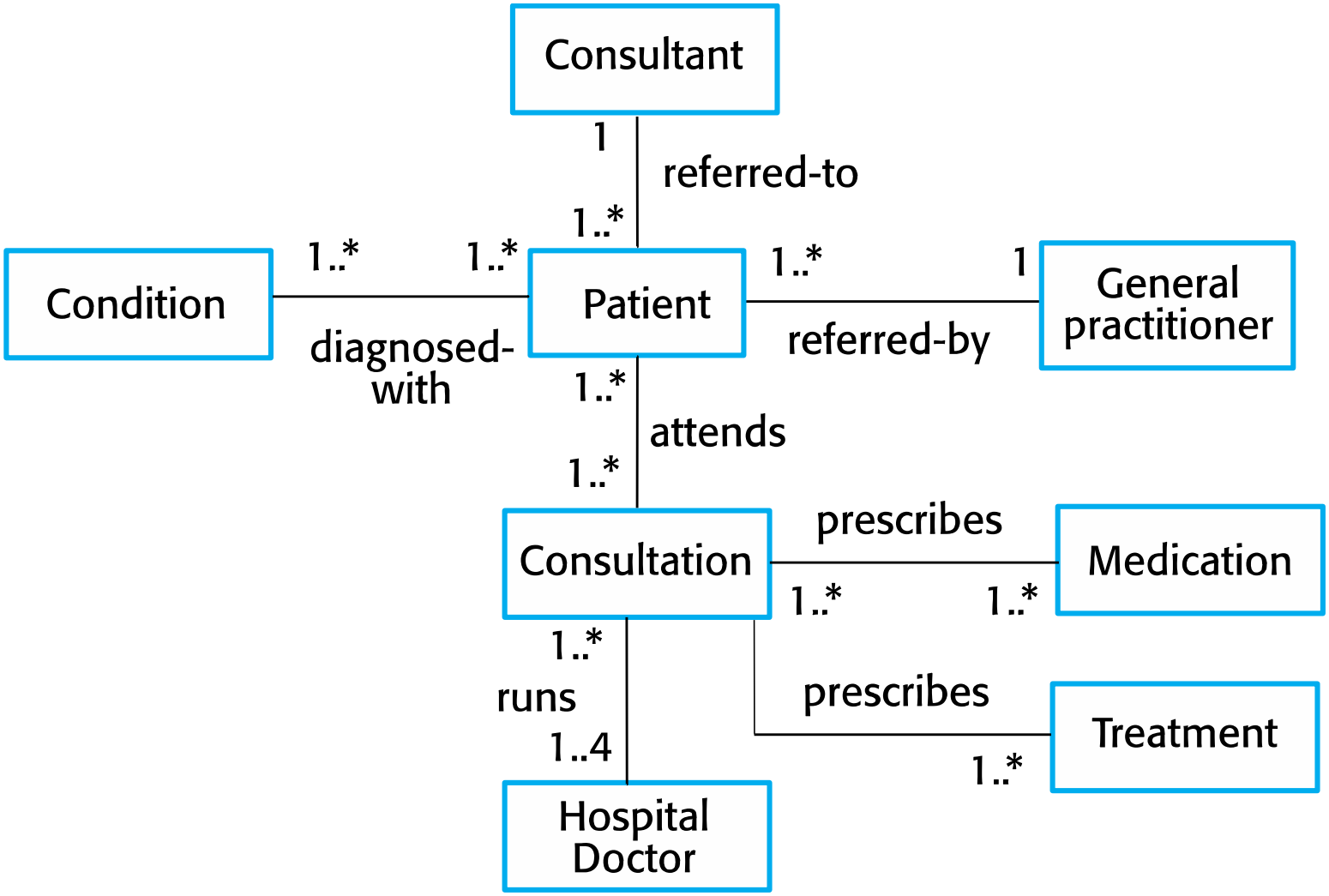
UML classes and association



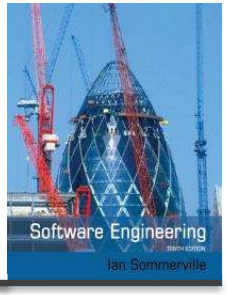


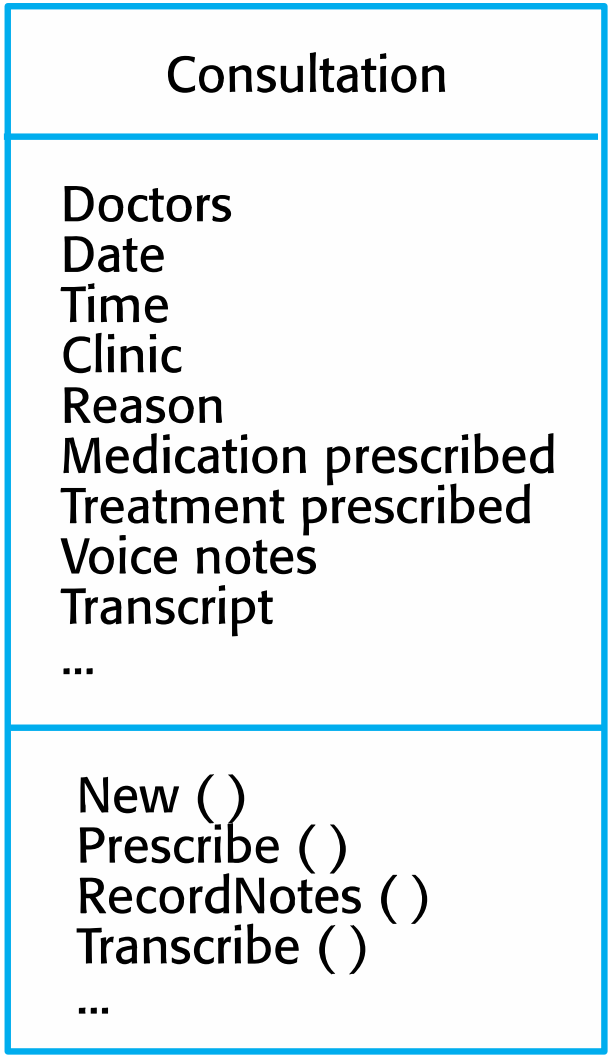
Classes and associations in the MHC-PMS



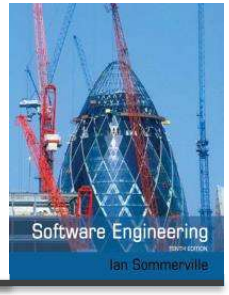


The Consultation class



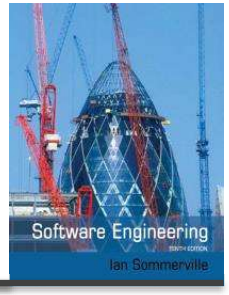


Generalization



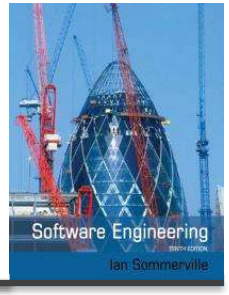
- Generalization is an everyday technique that we use to manage complexity.
- Rather than learn the detailed characteristics of every entity that we experience, we place these entities in more general classes (animals, cars, houses, etc.) and learn the characteristics of these classes.
- This allows us to infer that different members of these classes have some common characteristics e.g. squirrels and rats are rodents.

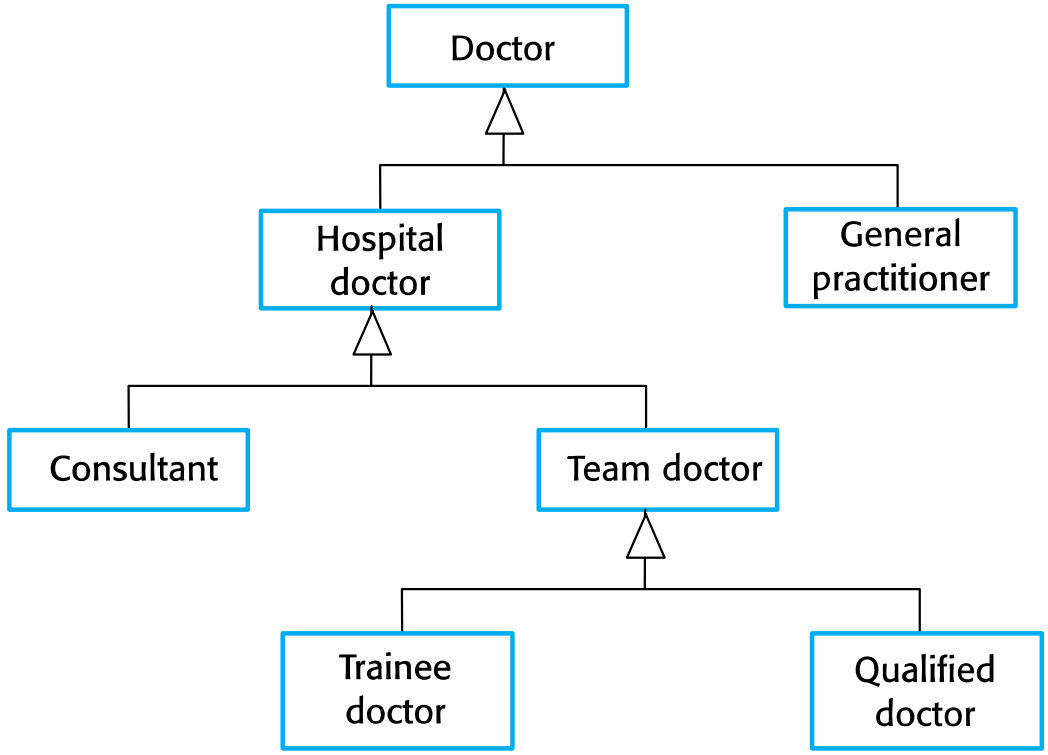
Generalization



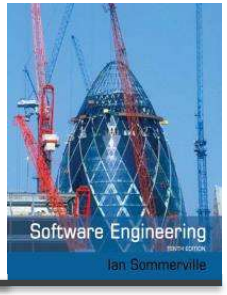
- In modeling systems, it is often useful to examine the classes in a system to see if there is scope for generalization. If changes are proposed, then you do not have to look at all classes in the system to see if they are affected by the change.
- In object-oriented languages, such as Java, generalization is implemented using the class inheritance mechanisms built into the language.
- In a generalization, the attributes and operations associated with higher-level classes are also associated with the lower-level classes.
- The lower-level classes are subclasses inherit the attributes and operations from their superclasses. These lower-level classes then add more specific attributes and operations.

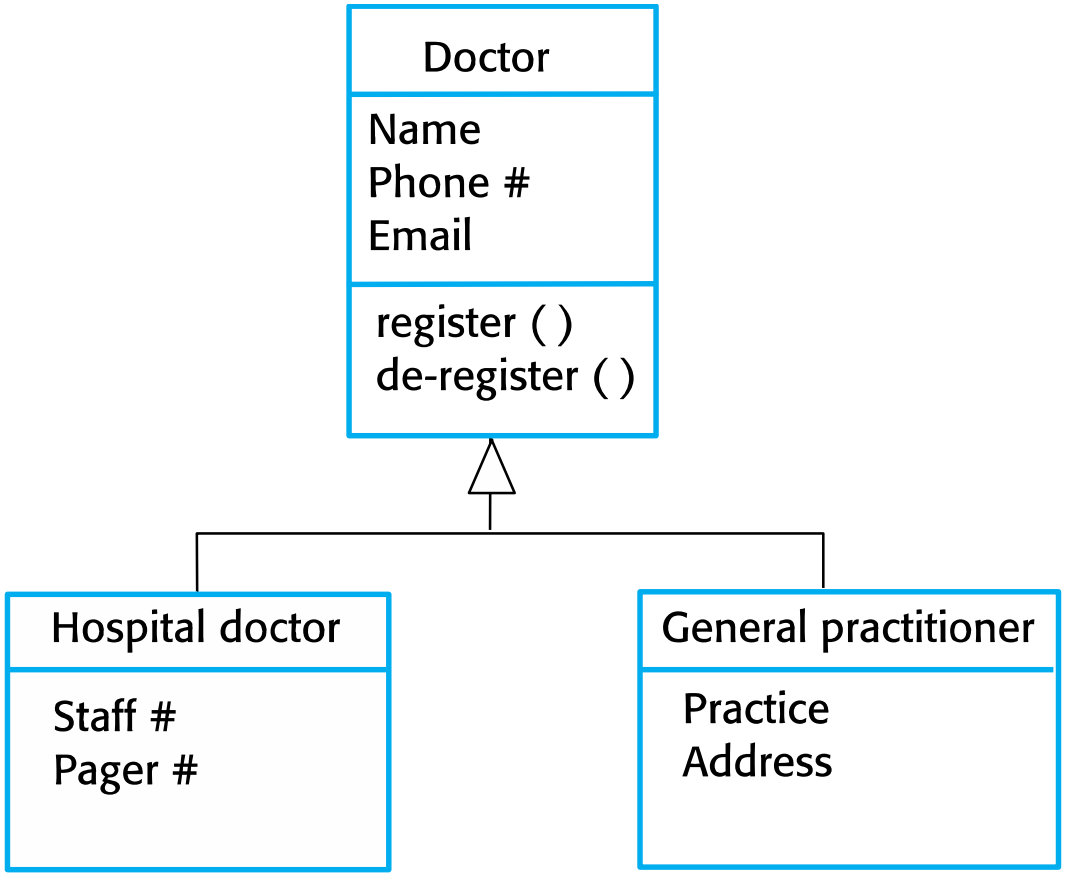
A generalization hierarchy



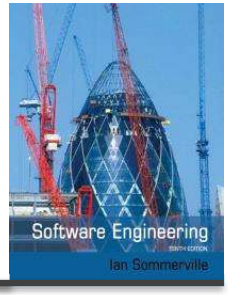


A generalization hierarchy with added detail



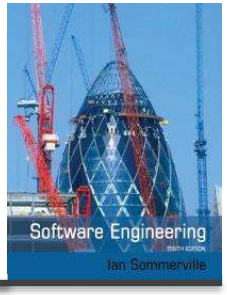


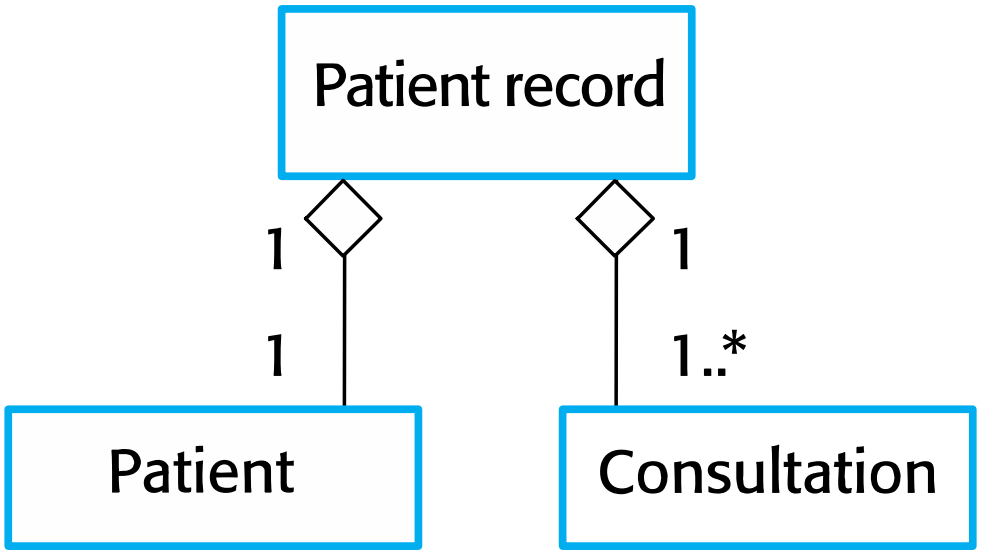
Object class aggregation models



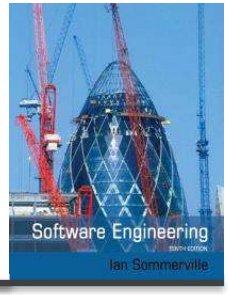
- An aggregation model shows how classes that are collections are composed of other classes.
- Aggregation models are similar to the part-of relationship in semantic data models.

The aggregation association

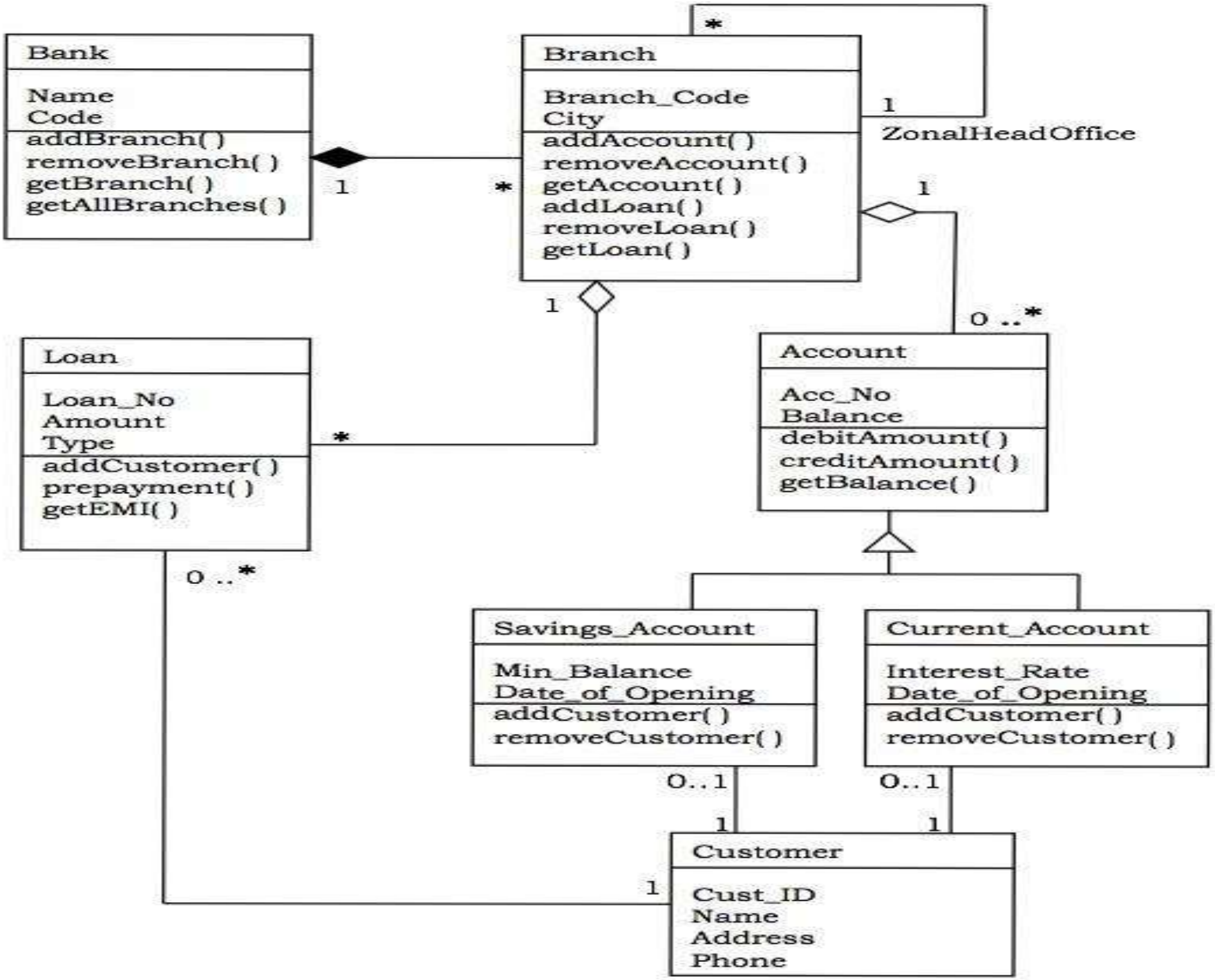




Structural Model Using A Class Diagram For A Banking System



A bank has many branches. In each zone, one branch is designated as the zonal head office that supervises the other branches in that zone. Each branch can have multiple accounts and loans. An account may be either a savings account or a current account. A customer may open both a savings account and a current account. However, a customer must not have more than one savings account or current account. A customer may also procure loans from the bank.


Bank
Name
Code
addBranch( )
removeBranch( )
getBranch( )
getAllBranches( )
Branch
Branch_Code
City
addAccount( )
removeAccount( )
getAccount( )
addLoan( )
removeLoan( )
getLoan( )
ZonalHeadOffice
Loan
Loan_No
Amount
Type
addCustomer( )
prepayment( )
getEMI( )
Account
Acc_No
Balance
debitAmount( )
creditAmount( )
getBalance( )
Savings_Account
Min_Balance
Date_of_Opening
addCustomer( )
removeCustomer( )
Current_Account
Interest_Rate
Date_of_Opening
addCustomer( )
removeCustomer( )
Customer
Cust_ID
Name
Address
Phone
1
*
0 ..*
0..1