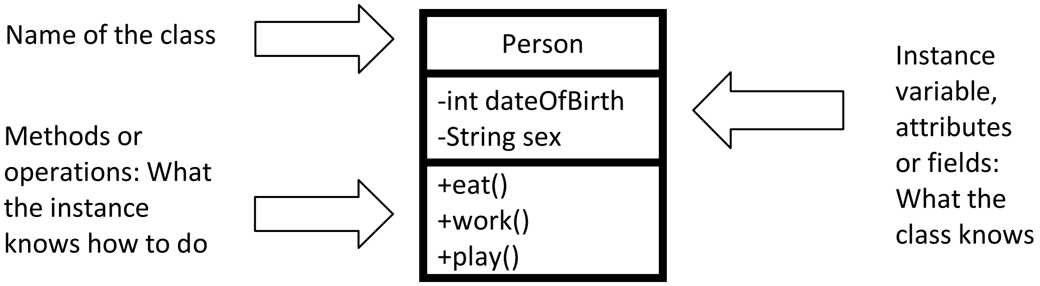
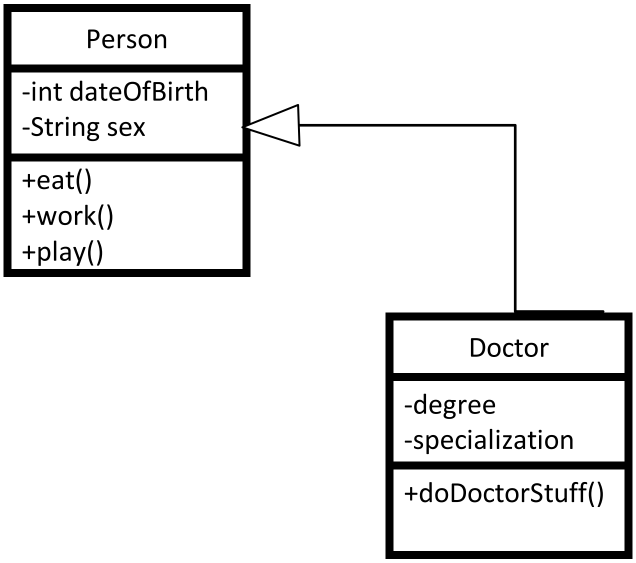
Recitation Guide for April 7th, 2008

I. Housing Keeping
   a. Homework 7 – due Wednesday April 9th
   b. Exam 2 – in class Friday April 11th
   c. Quiz 4 – in class Friday April 18th
   d. Quiz 3 solution posted to T-square. Grades and papers should be returned the latest on Wednesday. Two weeks from Wednesday for regrades. Regrades to Dawn.

II. UML (Unified Modeling Language)
   a. Individual diagram


Name of the class
Methods or operations: What the instance knows how to do
Person
-int dateOfBirth
-String sex
+eat()
+work()
+play()
Instance variable, attributes or fields: What the class knows

   b. Specialization/Inheritance relationship



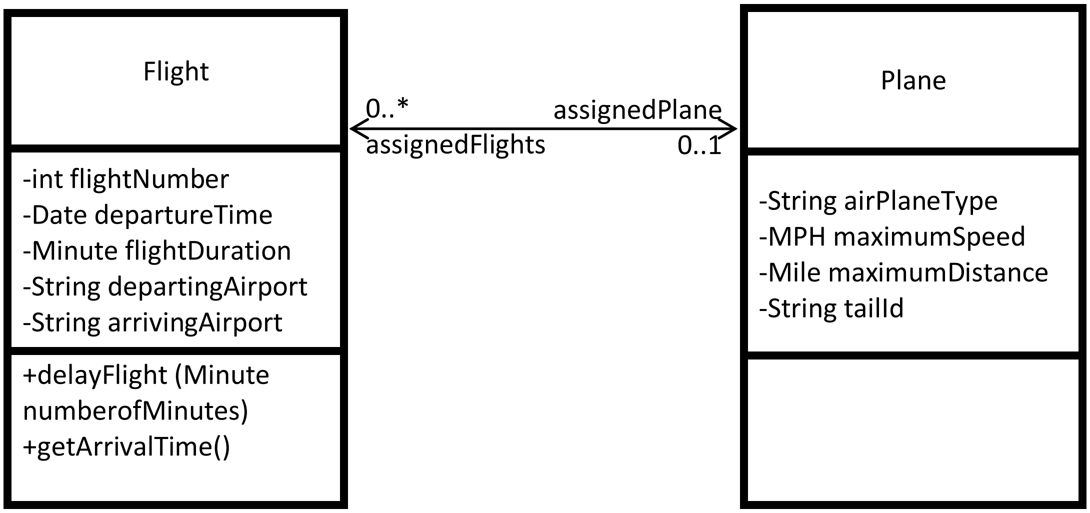
Doctor is a subclass of Person. Therefore Doctor is a specialization of Person.

c. Visiblity

Visibility Table

Mark	Visibility
+	Public
#	Protected
-	Private

d. Bi-directional relationship



A bi-directional association is indicated by a solid line (or a line ending in arrows on either side) between the two classes. At either end of the line, you place a role name and a multiplicity value. The Flight is associated with a specific Plane, and the Flight class knows about this association. The Plane takes on the role of "assignedPlane" in this association because the role name next to the Plane class says so. The multiplicity value next to the Plane class of 0..1 means that when an instance of a Flight exists, it can either have one instance of a Plane associated with it or no Planes associated with it (i.e., maybe a plane has not yet been assigned). The figure also shows that a Plane knows about its association with the Flight class. In this association, the Flight takes on the role of "assignedFlights"; the diagram tells us that the Plane instance can be associated either with no flights (e.g., it's a brand new plane) or with up to an infinite number of flights (e.g., the plane has been in commission for the last five years).

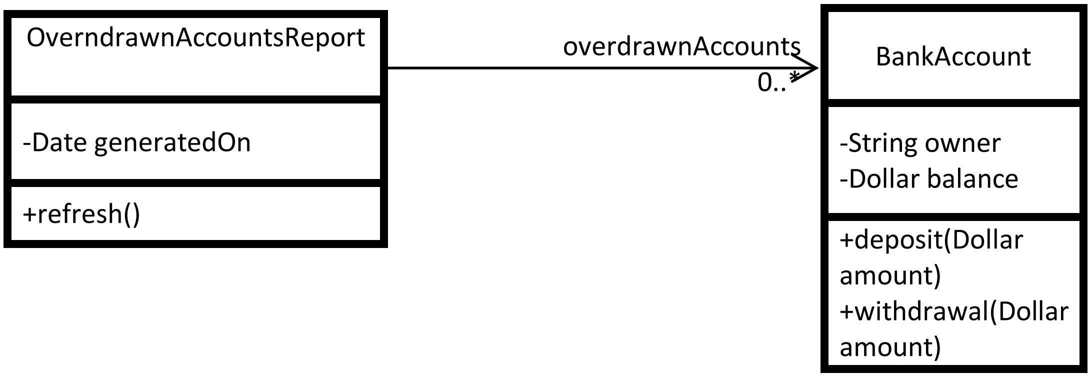
e. Multiplicity

Multiplicity Table

Multiplicity value	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
*	Zero or more
1..*	One or more
3	Three only
0..5	Zero to Five
5..15	Five to Fifteen

f. Uni-directional relationship

In a uni-directional association, two classes are related, but only one class knows that the relationship exists.



A uni-directional association is drawn as a solid line with an open arrowhead (not the closed arrowhead, or triangle, used to indicate inheritance) pointing to the known class. Like standard associations, the uni-directional association includes a role name and a multiplicity value, but unlike the standard bi-directional association, the uni-directional association only contains the role name and multiplicity value for the known class. In our example in the figure, the OverdrawnAccountsReport knows about the BankAccount class, and the BankAccount class plays the role of "overdrawnAccounts." However, unlike a standard association, the BankAccount class has no idea that it is associated with the OverdrawnAccountsReport.

g. For more information see:
http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/

III. DES (Discrete Event Simulations)

  a. Remember that:

    i. In discrete simulations not every moment in time is simulated.

    ii. Queues are FIFO (First in, First out) structures.

  b. The EventQueue

    i. Because there is no time loop, time events need to be placed in the **EventQueue**, where the first one removed (processed) is the event that occurs the earliest in time. At each run step, the next scheduled event with the lowest time gets processed. The current time is then that time, the time that that event is supposed to occur.

  c. Agents

    i. In discrete event simulations, agents do not just act. Instead, the agents wait for events to occur and schedule new events to correspond to the next thing that they are going to do. Events also get scheduled according to different probabilities.

    ii. However, agents cannot always do what they want to do because of limited resources. An agent is **blocked** until more resource is available.

IV. Continuous Simulations

  a. Remember that:

    i. Continuous simulations are driven by time.

  b. Each time step, the agents are told to act. The interactions between agents and the environment are based on their coded behaviors.

V. Go over Homework 7.

VI. Go over Quiz 3.