# Assignment #4

Practice with Object-oriented analysis & design, and GRASP

CS 4354 Summer II 2014
Instructor: Jill Seaman

**Due:** at the beginning of class **Monday, 7/28/2014**
Submit a "hard copy" (probably hand-written, optionally computer-generated) only.
Do this assignment with your partner and submit one copy with both names on it.

---

**Problem Domain:**

- The problem domain is a library management system.
- Books are checked out, checked in and reserved (put on hold) by library members.
- Students may check out books for 4 weeks, and faculty for 3 months.
- Library members are fined $0.25 per day that the books are overdue to a maximum of $5.00 per overdue item.
- The library also has other resources that can be checked out, including music CDs, software and videos. These resources may only be checked out for one week at a time. However, the overdue fines are the same as they are for books.
- Members may have no more than 10 resources checked out to them at any point in time.
- Members may not check out any additional resources if they have any unpaid fines.
- Any checkable library resource may be renewed as long as no other library member has requested it.
- Library members can browse the catalogue of resources to determine their status e.g. on the shelf, reference, out on loan, reserved etc.

**1** Identify at least four primary use cases and related actors implied by the Problem Domain description. Draw the results in a UML use case diagram

**2A** Write the textual description for the **checkout resources** use case, including Participating Actors, Flow of Events, Exceptional Flow of Events, and Entry and Exit Conditions. You may need to include things not mentioned specifically in the Problem Domain description.

**2B** Identify the entity, boundary, and control objects from your use case. Be sure to establish a control object for the use case.

**2C** Draw a sequence diagram that describes the checkout resources use case. Annotate as many messages as possible with a note indicating the GRASP (Expert, Creator, and so on) and/or any other pattern or reason that justifies it.

**2D** Draw a class diagram capturing the classes involved in the checkout resources use case.  Include associations, attributes, and operations.  Note that you may need classes and attributes for things not mentioned in the Problem Domain description.

**3A-D** Repeat steps A through D with another one of your use cases.  It should involve at least 2 entity objects and the sequence diagram should have 3 or more messages.