# Design Problems

The following problems provide design scenarios for you to model in Java classes. Read each scenario and then answer the questions below.

This page is left empty so that you can open this packet and see the problem description and questions at the same time.

1. Imagine a shipping company that only ships packages to certain predefined locations (could be intermediate shipping centers, could be holding centers where customers stop by to get their packages).  As packages move, they need to keep a running record of every location they've visited, the time they arrived there, and who “signed off” that the package arrived in good condition.  
     
   When a delivery truck arrives at a location with packages, the unloaders record the arrival of each package in through a simple web interface.  This does two things: (a) it updates the list of all packages at that location, and (b) it updates the package history of the package to reflect its new position.  
     
   All user interaction is handled by a class called PackageSystem which also includes the main.  This class includes a method:  
     
   registerPackageArrival(packageId, locationId, signoffPersonName)

See next page for questions to answer.

A) Draw a UML class diagram showing how you would design this system. You do not need to include every method or field - just the important ones.  It should be clear from your diagram how all the data mentioned above is stored, and it should follow the OO principles we've discussed in class.

B) Describe in a few sentences how when the registerPackageArrival function is called, the package's record is updated and the location's record is updated.  Make it clear what methods call what other methods, and reference what you wrote in your UML diagram.

1. Imagine a chain restaurant that sells pre-made frozen meals.  The chain has several retail outlets.  Each outlet has the same variety of meals, though they may have different number of each meal in stock. The system includes a meal-id which is unique, the published name of the meal (e.g. "Turkey Dinner for One"), and the price of the meal which is the same at each outlet.  Each retail outlet's inventory needs to be tracked so the central factory can send more meals when inventory gets low.  
     
   When a customer buys something, a record must be kept of the name of the customer who bought the meal and the store's inventory needs to be modified to reflect whatever has been removed.  
     
   All user interaction is handled by a class called MealSystem which also includes the main.  Here is a method which is part of meal system in the MealSystem class:  
     
   recordMealSold(customerName, mealId, retailOutletId)  
     
   See next page for questions to answer.

A) Make a UML diagram for this system.  You do not need to include every method or field - just the important ones.  It should be clear from your diagram how all the data mentioned above is stored, and it should follow the OO principles we've discussed in class.

B) Describe in a few sentences how the recordMealSold command is implemented, referencing your UML diagram.  Be sure to note what methods are called by what other methods, and how the fields of your classes are updated.

1. Imagine an auction system that auctions a small number of items each day.  Each item auctioned has an id number and a brief description. Throughout the day, users can stop by and view the items available today and decide to bid a specific amount for each item they want to bid on; they can't see what others have bid.  They system stores the users name, an id, and any outstanding balance they might have for unpaid items they have won.  
     
   Then, at the end of the day the owner enters the ``end all auctions'' command into the system.  For all auctions, the user with the current highest bid for that item wins the auction.  Then, the next day, the users can come by, see the items they won, and pay for their bids.  
     
   All user interaction is handled by a class called AuctionSystem which also includes the main.  Here are four methods that are part of AuctionSystem:  
     
   addAuction(itemId, description) - starts a new auction for a particular item  
     
   registerBid(itemId, userId, bidAmount) - registers a bid for a particular item in the system.  
     
   endAllAuctions() - causes all current auctions to end.  The current high bidder for each item currently being auctioned wins and the items are no longer available for future bids.  
     
   printWonItems(userId) prints out the name of all the items user has won  
     
   See next page for questions to answer.

A) Make a UML diagram for this system, playing particular attention to the fields and methods needed by the four above methods.  You do not need to include every method or field - just the important ones.  It should be clear from your diagram how all the data mentioned above is stored, and it should follow the OO principles we've discussed in class.

B) Describe in a few sentences how the endAllAuctions() command is implemented, referencing your UML diagram.  Be sure to note what methods are called by what other methods, and how the fields of your classes are updated.