

This year (2008-2009) the continuous assessment element of CA422 consists of two separate parts (I and II), each worth 10% of the overall marks for the module. The overall mark for CA422 will be computed as the average of the end of semester exam mark (weight of 0.8) and the continuous assessment total mark (weight of 0.2).

NB: The continuous assessment is to be done by each student individually (NOT in teams). University rules regarding plagiarism and copying will be strictly applied.

Part I of continuous assessment (10%):

I.1 "A hotel has a number of rooms that are rented out for meetings. Rooms are rented for a week at a time. Assume weeks are designated from 1 to 52. A room is allocated for at most one meeting. Rooms have a maximum capacity. A room is either allocated or free. Each meeting is represented by a unique identifier (a number) when a booking is made. Each booking records the number of delegates who will attend the meeting. The booking is rejected if the number of delegates exceeds the capacity of the room. When a booking is cancelled the booking is deleted and the room (or rooms) are free for subsequent bookings."

(a) Create a Z state space for the room booking system together with a definition of valid system responses.

(Hint: response ::= nomorerooms | booked | deleted etc.)

(b) Specify an operation **book**, to book a room for a meeting scheduled to happen in a particular week (1 ..52).

(c) Specify an operation **cancel**, to cancel a room booking for a meeting.

(d) Specify an operation to return a list of all meetings currently booked, with a delegate count greater than some given number.

(e) Specify an operation to return the number of free (unbooked) rooms for a particular week.

(f) Specify an operation to list the number of room bookings in the least booked week of the year.

Notes:

a) A booking should include the identity of who has made it.

b) Error handling schemas should be defined to handle where pre-conditions are violated.

I.2 Extend your solution to **I.1** to cover the case of bookings over an entire hotel chain. It should be possible to book a meeting room in any member hotel of the chain. You should indicate any modifications needed in your overall specification and in the schemas for the individual operations. In addition, you should specify new operations to allow the addition of a hotel to the chain and the disposal of a chain (for which a pre-condition should be that there are no rooms booked for the hotel to be disposed of).

Part II of continuous assessment (10%):

II.1 Formulate a Z specification for a single-server queue system, such as in a small shop with one purchase point. Your specification should distinguish between being in the queue and being served. You should include specification of operations for a customer to join the queue, to enter service, and to leave service. Also, you should specify an enquiry operation to output the queue length (i.e. the number of customers in it). For this part of the assignment assume that the next customer to enter service is chosen randomly from those in the queue (e.g. you could have a variable "chosen?" whose value is "yes" for the chosen customer).

II.2 Extend the specification of **II.1** such that the queue discipline is first-come first-served, making any appropriate changes to your solution for part **II.1**. Also, extend **II.1** to include "balking" (i.e. a potential customer will not join the queue if its length is greater than some global limit).

II.3 Extend the specification of **II.2** to the case of several servers each with their own queue (such as in a supermarket). Make any necessary additions and modifications to the existing single-server specification. In particular, specify that a customer will join the shortest available queue. Also, include operations to add a new server and to remove an existing one.