UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2003

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BSc Honours Degree in Mathematics and Computer Science Part II
MSci Honours Degree in Mathematics and Computer Science Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C220=MC220

SOFTWARE ENGINEERING - DESIGN I

Wednesday 7 May 2003, 15:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

Section A *(Use a separate answer book for this Section)*

1 Throughout this question state clearly any additional assumptions made.

a i) Explain why low-cohesion and high-coupling are undesirable in software design?

ii) Give three examples of guidelines that can be used in assigning responsibilities to classes in order to reduce coupling and increase cohesion.

b SWD is a software consultancy company that undertakes systems development contracts for various clients. Each contract has a start date and an end date. SWD employs Analysts, Architects and Project Managers which are allocated to the various contracts for a duration which may be less than the duration of the contract. Duration is specified as a start date and an end date. Contracts can be software design contracts, implementation contracts, or systems development contracts. Implementation contracts are outsourced to an external provider and systems development contracts may comprise several subcontracts. Each employee has a daily rate which determines how much SWD should invoice clients for their work. Assume that an employee cannot be simultaneously assigned to two contracts. SWD needs to design a system to manage employees and contracts. In this system a *ContractManager* class maintains references to all the contracts while a *HRManager* class maintains references to all the employees. The system must be able to: *i)* determine which employees are free between a given start date and end date and *ii)* periodically generate invoices for those contracts where the end date has passed. The invoices must specify the names of the employees, the period (start date and end date) during which they have worked on the contract and the total cost of their involvement.

i) Draw a class diagram showing a possible software design for the system described above, including all the necessary attributes and methods.

ii) Briefly explain how the following tasks would be implemented in your system, showing which classes would be responsible for which tasks and why this responsibility would be attributed to them.

- Creating an instance of a contract.
- Identifying the period for which a specific employee has worked on a specific contract.
- Listing all the employees currently working on a particular contract
- Printing all the invoices

c When a new contract is created, the system must find a Project Manager available for the entire duration of the contract and assign him/her to the contract for the entire duration of the contract.

i) Draw a collaboration diagram indicating the invocations which occur in the system. Indicate any transient links.

ii) Briefly discuss any alternatives.

The three parts a, b &c carry, respectively, 20%, 50%, 30% of the marks.

2a UML defines both sequence and collaboration diagrams for representing interactions between objects. Giving examples for each type of diagram, briefly explain what these diagrams represent, how they differ and why they are both useful during the software design process.

b The following is a description of the functioning of an ATM cash machine from which customers can withdraw money or check the balance of their account.

The cash machine can be switched on and off. When no customer is being served the cash machine continuously displays a main screen. When a customer inserts a card, if the card is readable, the information is read from the card and the customer is asked to input his/her password (PIN). If the card is not readable, the card is ejected and the customer is asked to withdraw it. Any cards that are ejected and not removed within 20 seconds are confiscated by the machine. When asked by the cash machine, the customer digits his/her PIN and presses the *Enter* key. This causes the PIN to be read, and the cash machine attempts to verify the validity of the PIN. If the PIN is not valid, then the customer is given two more opportunities to enter it. At the third consecutive unsuccessful attempt the card is ejected. Assume a counter is available.

If the PIN is valid the customer can choose to make a withdrawal or request the balance of the account. If the customer chooses to make a withdrawal, then he/she is asked to enter the amount. When the *Enter* key is pressed the amount is read and authorisation is sought from the bank. If the withdrawal is authorised, the amount requested is dispensed. Otherwise, the customer is prompted to make a new operation (i.e., withdrawal or balance enquiry). The card is ejected when the customer removes the money dispensed. If the customer fails to remove the money within 20 seconds both the card and the money are confiscated. When the customer requests the balance of the account, the balance is requested from the bank and then displayed to the customer. By pressing *Enter* the customer can make a new operation. While performing any operation, the customer can press *Exit*. In this case the card will be ejected.

Any failure by the customer to select an option or enter an input within 20 seconds will result in the card being ejected.

Draw a state chart modelling the behaviour of the ATM cash machine. State explicitly any additional assumptions that you make.

c i) Explain which composite states would be desirable in your answer to b) and why.

ii) Assume that an action `notify(<event type>)` can be used to generate an event. Modify your answer to part b) to show a counter that counts the unsuccessful attempts to enter the PIN working concurrently with the attempts to validate the PIN. Show only the states and transitions that are affected.

The three parts a, b &c carry, respectively, 20%, 60%, 20% of the marks.

3a Briefly state the purpose of the "Chain of Responsibility" design pattern and explain how it works using both class and collaboration diagrams.

b In an on-line discussion system, several participants may join different chat rooms. When a participant sends a message to a chat room, all the participants receive the message and the message is appended to the chat room's log. Participants and chat rooms are identified by their names. Names and messages are implemented as Strings.

You can assume that the following constructs exist.

```
interface Collection {
    boolean add(Object o) ;
    boolean remove(Object o);
    boolean contains(Object o);
    boolean isEmpty();
    Iterator iterator();
    int size();
}
```

```
interface Iterator {
    boolean   hasNext();
    Object    next();
    void      remove();
}

class LinkedList
implements Collection { …}
```

i) Identify which design pattern would be most appropriate for implementing this.

ii) Define the interfaces that chat rooms and participants must implement.

iii) Give the Java implementation of the chat room and participant classes. Assume that messages received by a participant are printed on the screen.

c There is a single list of available chat rooms which can be obtained by the participants. When created, a chat room appends itself to this list. Modify your implementation from question b) to provide this list and to provide the implementation of the following functions:

```
class Participant { …
    public void join(String c)     // join all chat rooms named c
    public void send(String c, String mes)
                    // send message mes to all chatrooms named c
}
```

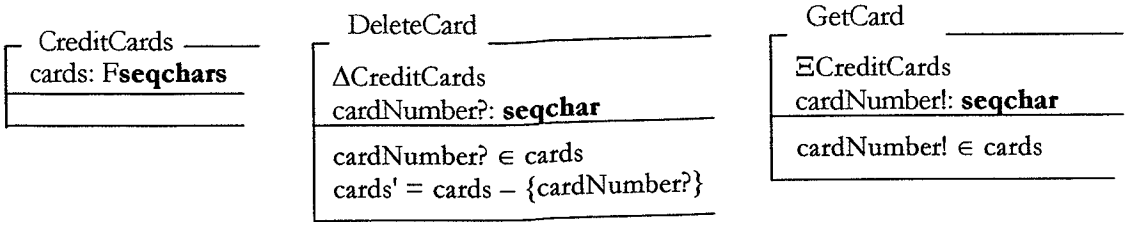
Only show those parts of the implementation that differ or are in addition to your answer to b).

The three parts a, b& c carry, respectively, 20%, 50%, 30% of the marks.

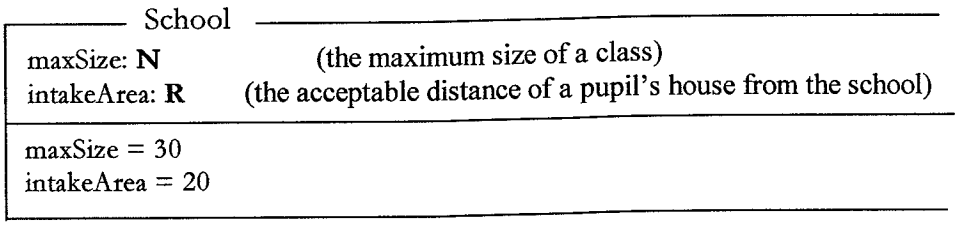
***Section B** (Use a separate answer book for this Section)*

4a i) Describe the standard usage of the four symbols ?, !, Δ, and Ξ in specifying operation schemas, and explain what is meant by *decorated inclusion* of one schema in another.

ii) Consider the following schemas. Write the two operation schemas *Request1* = *GetCard* ∧ *DeleteCard* and *Request2* = *GetCard* ∨ *DeleteCard* in full, without using any schema inclusion. State whether these two operations are consistent or not, and explain your answer.



b A primary school for pupils from age 5 to 11 uses an automatic system for handling the student registration process. The school has a maximum class size of 30 pupils. The system keeps track of the size of each class and the names of the pupils that are in each class. The intake process of 5 years old pupils is constrained by the following two criteria. Pupils who are not siblings of existing pupils are accepted only if their house location is within the intake area of the school. Pupils who are siblings of existing pupils are accepted independently of their house location. Pupils older than 5 years are accepted if and only if there is space in their class. The following schema *School* defines the basic parameters of the system.



i) Write a state schema *SchoolState* that includes the schema *School* and two functions *classSize* and *classPupils*, which specify respectively the size of each class and the set of pupils names in each class. Assume a predefined type *ageGroup* = [5,..,11]. Include relevant constraints on these functions.

ii) Write operation schemas for the following operations using a defensive approach. All the schemas include the input *pupilAge?* (of type *ageGroup*), *pupilName?* and the output *report!*. Marks will be awarded for correct use of Δ and Ξ.

IntakeFirstYear, which specifies the registration process of 5 years old pupils. The operation uses the additional input *sibling?* (of type **Boolean**), which says if the pupil is a sibling of an existing pupil or not, and *location?* (of type **R**), which specifies the distance of the pupil's house from school. It returns *OK*, when the pupil is accepted, "Outside intake area", when the pupil's house is not within the intake area, and "Class full", when the class has already reached its maximum size.

IntakeFurtherYear, which specifies the registration of a pupil older than 5 years.

iii) Consider the operation *Intake* = *IntakeFirstYear* ∨ *IntakeFurtherYear*. Is this operation total? Explain your answer.

The two parts carry, respectively, 30% and 70% of the marks.