UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2002

MSc in Computing Science
for Internal Students of the Imperial College of Science, Technology and Medicine

PAPER M1

PROGRAM DESIGN AND LOGIC

Monday 29 April 2002, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

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

1a i) Show $\neg(A\vee B) \vdash \neg A \wedge \neg B$ using syntactic techniques only. At each stage of the proof clearly state the inference rule and the wffs used.

ii) Show $A\vee B\rightarrow C \equiv (A\rightarrow C)\wedge(B\rightarrow C)$ using any syntactic or semantic techniques. At each stage of the proof clearly state the inference rule or equivalence and the wffs used.

iii) Show $(A\wedge B)\vee\neg C\rightarrow(E\rightarrow F) \vdash A\wedge\neg F\rightarrow(E\rightarrow C\wedge\neg B)$ using only syntactic techniques and, if required, (i) and (ii), above. At each stage of the proof clearly state the inference rule or (i) or (ii) and the wffs used.

b Consider the action of driving a car from one location to another. Let the following be a formalisation of its preconditions in the situation calculus:

$$\forall X,Y,L1,L2,S \ (\text{poss}(\text{drive}(X,Y,L1,L2), S) \leftrightarrow \text{in}(X,Y,S) \wedge \text{at}(X,L1,S) \wedge \text{at}(Y,L1,S) \wedge \text{has}(X,\text{igkey}(y),S))$$

where the predicates have the following meanings:

poss(drive(X,Y,L1,L2), S):	in state S it is possible for X to drive car Y from location L1 to location L2,
in(X,Y,S):	in state S X is in car Y,
at(X,L,S):	in state S X is at location L,
has(X,igkey(y),S):	in state S X has the ignition key to car Y.

So the logic sentence above states that it is possible for X to drive car Y from location L1 to location L2 in state S if and only if in that state S X in car Y, both X and Y are at location L1 and X has the ignition key to Y.

Assume that locations are postal codes, for example SW7, SW15, etc.

i) Considering the formalisation above and your own common sense understanding of the act of driving, being at a location, and being in a car, for each of the following two sentences first state whether it is true or false and then give a brief justification for your answer.

(1) $\forall X,L1,L2,S \ (\text{at}(X,L1,S) \wedge \text{at}(X,L2,S) \rightarrow L1=L2)$

(2) $\forall X,Y,L1,L2,S$
$(\neg\text{poss}(\text{drive}(X,Y,L1,L2), S) \rightarrow \neg \text{at}(X,L1,S))$

ii) Express three of the postconditions of the act of a person driving a car from one location to another in the situation calculus using the predicates listed above. Do not forget that in the situation calculus one state changes into another by one action only.

Parts a and b each carry 50% of the marks.

2a The sentence “A soldier visited every cave” is ambiguous, i.e. it has more than one interpretation.

i) Using predicate logic and the following predicates formalise two different interpretations of the sentence.

cave(X)	to mean X is a cave,
soldier(X)	to mean X is soldier
visited(X,Y)	to mean X visited Y.

ii) One interpretation implies the other. State which one of your logic sentences, given in answer to (i), implies the other.

iii) Prove the implication you indicated in (ii) using syntactic techniques only. At each step of the proof state the inference rule and the wffs used.

b The following is a definition of the relation

front_of(F,L) : F is a non-empty front end sublist of the list L

using the append/3 relation.

```
front_of(F,L) :- append(F,B,L),F≠[].

append([],Y,Y)
append([U|X],Y,[U|Z]) :- append(X,Y,Z).
```

i) Give an alternative two clause recursive definition of the front_of/2 relation that does not use append/2 or any other relation.

ii) Construct the Prolog search tree for the query:

front_to(X,[1,2])

using either the given program of (b), or your answer to (b).

Parts a and b each carry 50% of the marks.

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

3 Consider the following simplified description of mortgages:

- Freehold properties have a value and an area.
- Leasehold properties have a value, an area, and a date of expiry of the lease.
- Mortgages may be taken on properties and have an outstanding amount. The outstanding amount of a mortgage on a freehold property may never exceed the value of the property divided by its area, whereas the outstanding amount of a mortgage on a leasehold property may never exceed the value of the property multiplied by the years remaining till the expiry of the lease, divided by its area.
- Mortgages are given either by banks or by building societies.
- Mortgages require monthly payments. The monthly payment of a mortgage held by a bank is the outstanding amount minus the mortgage relief, multiplied by the sum of the bank's interest rate and the interbank rate. The monthly payment of a mortgage held by a building society is the outstanding amount minus the mortgage relief, multiplied by the building society's interest rate.
- The mortgage relief is the same for all mortgages and may change with time.
- The interbank rate is the same for all banks, and may change with time. The interest rate may be different for each bank, and may change with time. The interest rate may be different for each building society, and may change with time.

a Develop a class diagram to describe the above. You may use UML or OMT. You will find a summary of the OMT notation on the 5th page of this exam paper.

b Write C++ classes (*i.e. declarations* and *function bodies*) to implement the above. You may use the class `Date` to represent dates:

```
class Date {
public:
        Date(int d, int m, int y)
        float operator -(Date)
        // returns number of years between receiver and argument
        static Date today()
        // returns today's date
};
```

c Write a test function where:

i) FH1 is a freehold property with area 50.00 and value 3,600,000. LH1 is a leasehold property with area 66.66, value 400,000 and an expiry date of 20th May 2070.

ii) The mortgage relief is 60.00, and the interbank rate is 3.00. B1 is a bank with interest rate 2.05.

iii) M1 is the mortgage obtained after applying to B1, for FH1 for an amount of 10,000 to be held by B1. Calculate the appropriate monthly payment.

iv) The mortgage relief changes to 50.00, the interbank rate changes to 1.00, and B1's interest rate changes to 3,70.

The three parts carry, respectively, 35%, 50%, 15% of the marks.

4 Consider the following simplified description of bets on football matches.

- The outcome of a football match is initially unknown, and later it may be victory for the first team, or victory for the second team, or a draw.
- Customers may open an account with a broker, and pay some money into the account. They may not open more than one account with the same broker.
- Two customers may apply to a broker to register their bets. A bet has an amount, it depends on a match, and contains the predicted outcome (i.e. first team to win, or second team to win, or draw), and a floating point number unequal 0 to represent the odds. A bet can be registered, only if the outcome of the match is yet unknown, and if both customers have an account with that broker, and each of them has in their account enough money to pay for this and all other outstanding bets.
- Brokers periodically clear their books by considering each bet registered with them whose outcome is known. If the outcome is the same as the one predicted, then 95% of the amount multiplied by the odds is transferred from the second customer's account to the first customer's account, and 5% of amount multiplied by the odds is transferred from the second customer's account to the broker. If the outcome is not the same as the one predicted, then 95% of the amount multiplied by the reciprocal of the odds is transferred from the first customer's account to the second customer's account, and 5% of the amount multiplied by the reciprocal of the odds is transferred from the first customer's account to the broker.

a Develop an class diagram to describe the above. You may use UML or OMT. You will find a summary of the OMT notation on the 5th page of this exam paper.

b Write C++ class declarations (*i.e. no function bodies*) to support the above.

c Write a test function, where

i) M1 and M2 are matches whose outcome has not yet been decided. BM1 and BM2 are brokers. C1 and C2 are customers. C1 opens an account with 3000 with BM1, and an account with 4000 with BM2, and C2 opens an account with 5000 with BM1.

ii) C1 and C2 apply to register a bet of 55.55 with BM1, where C1 predicts the outcome of M1 to be a draw with the odds 0.75.

iii) The outcome of M1 is that the first team won.

iv) BM1 clears his books.

You may want to use the following template classes

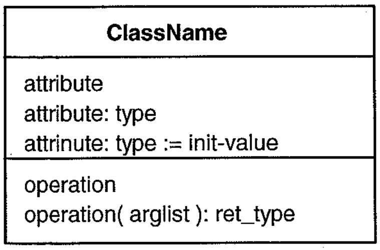
```
template <class T>
class List { // a list of Ts
public:
    List(); // an empty list
    void insert(T* aT); // inserts aT into the list
    T* get();    // returns an elemet, and removes it from list ...};

template <class key, class T>
class Table { // a table with entries of type T*
public:
    Table(); // an empty table
    void insert(key k,T* aT); // inserts aT under k
    T* get(key k);    // returns entry under k, if any ...};
```

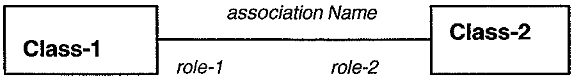
The three parts carry, respectively, 20%, 60%, 20% of the marks.

OMT: Basic Notation for Object Models

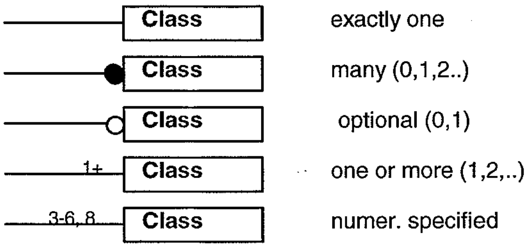
Class:



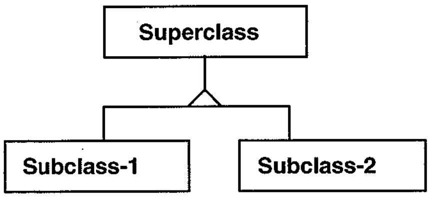
Association



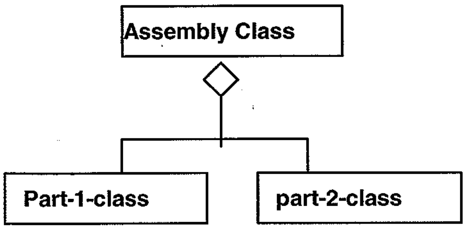
Multiplicity of Association/Aggregation



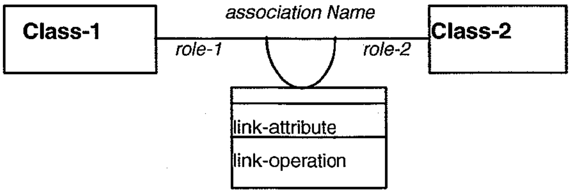
Generalization (Inheritance)



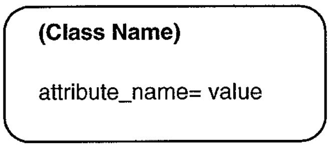
Aggregation



Link Attributes

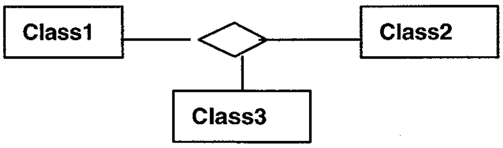


Object Instance



$ for class operations/attributes

Ternary Association


Class1
Class2
Class3