IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2014

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

PAPER M1

PROGRAM DESIGN AND LOGIC

Friday 9 May 2014, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

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

Note: All natural deduction proofs must be presented clearly, with wff numbering, where appropriate, indentations, and explanations. Marks will be deducted for unclear and poorly presented proofs. When using natural deduction you may use any of the primitive and derived rules.

1 a Formalise in predicate logic the sentences (i)-(iv), below, that talk about the activities of a school. Use only the predicates listed below. Ensure that you present your formulas clearly, using brackets to correctly identify the scope of quantifiers and disambiguate where necessary.

day(X)	*to mean X is a day class.*
eve(X)	*to mean X is an evening class.*
teaches(P, C)	*to mean person P teaches class C.*
salary(P, Scode)	*to mean person P is on salary code Scode.*
subject(Sub, C)	*to mean subject Sub is taught in class C.*
type(Sub, T)	*to mean subject Sub is of type T (for example subject Latin is of type languages).*
enrolled(S, C)	*to mean student S is enrolled on class C.*
sitsExam(S, Sub)	*to mean student S sits an examination on subject Sub.*

i) There are day and evening classes.

ii) All who teach a day class are on salary scheme *sd*, unless they also teach an evening class, in which case they are on salary scheme *sde*. Those who teach evening classes only are on salary scheme *se*.

iii) The subjects taught in the evening classes are (of type) languages and humanities. There are also day classes teaching these types of subjects. But Latin is not a subject taught in any day or evening class.

iv) Everyone who is enrolled on a day class will sit an examination on the subject of that class, but no one enrolled on evening classes only will sit any examinations.

b Imagine a retail store with sensors that track its merchandise. The store has a rule of logic for recognising when an item of merchandise is stolen, as follows:

> If an item of merchandise exits the store and it has not been paid for by the time it exits then it is stolen at the time of exiting the store.

Consider the following sentences S1-S4 of logic, assuming that *exit(I, T)*, *paid_for(I, T)*, *stolen(I, T)* mean, respectively, that item *I* exits the store at time *T*, item *I* is paid for by time *T*, and item *I* is stolen at time *T*.

S1) $\forall X\ \forall T\ (exit(X, T) \wedge \neg\ paid\_for(X, T) \rightarrow stolen(X, T))$

S2) $\forall X\ \forall T\ (\neg stolen(X, T) \rightarrow (\neg\ exit(X, T) \wedge paid\_for(X, T)))$

S3) $\forall X\ \forall T\ (paid\_for(X, T) \vee stolen(X, T) \rightarrow exit(X, T))$

S4) $\forall X\ \forall T\ (\neg stolen(X, T) \rightarrow (\neg\ exit(X, T) \vee paid\_for(X, T)))$

i) Two of these sentences formalise the store's rule correctly. Identify them and show that they are equivalent with one another, using equivalences only.

ii) For one of the sentences that does not represent the store's rule correctly find an equivalent sentence that has only $\exists$ as quantifier and $\neg$, $\wedge$ and $\vee$ as connectives.

iii) A more sophisticated version of the retail store's rule would be as follows:

If an item of merchandise exits the store and it has not been paid for by the time it exits then it is stolen at the time of exiting the store, unless its removal from the store has been approved by the time of removal by an authorised person. The only authorised persons are the manager and the deputy manager.

Represent this more sophisticated rule by sentences of predicate logic using the predicates given above and the following:

approved(Y, X, T)	*to mean removal of X is approved by Y by time T.*
authorised(X)	*to mean X is an authorised person.*
manager(X)	*to mean X is the manager.*
depManager(X)	*to mean X is the deputy manager.*

Parts a, b carry 55%, 45% of the marks, respectively.

2 a A DeMorgan rule is as follows:

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

Prove the equivalence by the application of the inference rules of natural deduction. Do not use any equivalences in your proof.

b Show that

$$S1, S2, S3, S4 \vdash \forall X\, (m(X) \wedge p(X) \rightarrow \exists Y\, t(X,Y))$$

using the inference rules of natural deduction and the equivalence above, if needed, where *S1, S2, S3, S4* are as follows:

S1	$\forall X\, (p(X) \rightarrow \neg(q(X) \wedge r(X)))$
S2	$\forall X\, (\exists Y\, s(X,Y) \rightarrow q(X))$
S3	$\forall X\, (m(X) \rightarrow \exists Y\, s(X,Y) \vee \exists Y\, t(X,Y))$
S4	$\forall X\, (\neg(n(X) \wedge r(X)) \rightarrow \exists Y\, t(X,Y))$

c Using the inference rules of natural deduction and (b) above, if needed, show that

$$S1, S2, S3, S4, S5 \vdash \forall X\, (\neg \exists Y\, u(X,Y) \rightarrow \neg\, (m(X) \wedge p(X)))$$

where *S1-S4* are as in (b) above and *S5* is as follows:

S5 $\quad \forall X\, \forall Y\, (t(X, Y) \rightarrow u(X,Y))$

Parts a, b, c, carry 25%, 50%, 25% of the marks, respectively.

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

3 Consider the following description of a hyperlocal messaging system which allows users to broadcast public messages to other users who are located nearby:

- Users have a position (x, y) where x and y are floating point numbers. Users also have a number of message credits.
- Users can broadcast a message from their current position p by specifying some message text t and a message radius r. The potential message recipients are those users located within distance r from p, excluding the sender. If the number of potential message recipients exceeds the number of message credits held by the sender, the message is not sent. Otherwise, message text t is delivered to the message recipients, and the number of message credits held by the sender is reduced by the number of message recipients.
- Users can update their positions and can also buy message credits.

You may assume the availability of the following template class:

```
template <class T>
class List {
  ...
public:
  void append(const T &item); // append item to list
  void remove(const T &item); // delete item from list
  int size(); // number of list items
  T *front(); // pointer to first list item
  T *next();  // pointer to subsequent list item(s)
};
```

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

b Write a test function where:

- User Alice is at position (3.5,0.5) and has 5 messaging credits. User Bob is at position (10.0,5.0) and has 0 messaging credits. User Charles is at position (7.0,7.0) and has 2 messaging credits.
- Alice broadcasts the message "Hello!" with a message radius of 10, and then moves to position (3.0,3.0).
- The number of message credits Alice has is displayed.
- Charles broadcasts the message "Come to my party at the Royal Albert Hall" with a message radius of 5.
- Bob buys 10 message credits.

c Write function bodies for your classes (excluding the template class).

The three parts carry, respectively, 35%, 30%, and 35% of the marks.

4 Consider the following description of Formula One racing cars.

- A racing car consists of a *chassis* and an *engine*, and is given a *model name*. An engine has a *mass*, a *power factor* and a number of *cylinders*. A chassis also has a mass. Each racing car is driven by a *driver*, who has a mass and a *name*.

- The *racing mass* of a car is the combined mass of the chassis, engine and driver. The regulations for Formula One state that this mass must not be less than 642kg. If the racing mass of the car is less then this, then ballast must be added to achieve this minimum weight.

- The amount of time T taken by a racing car to travel a given distance D is a function of the racing mass of the car M, the number of cylinders C and the power factor P of the engine. The times for cars containing *TurboCharge2000* and *SupermanV3* engines are given by, respectively, formulas (1) and (2) below:

$$(1) \quad T = \frac{M \times D \times (1 - \log(1 + e^{-D}))}{P \times C^2} \qquad (2) \quad T = \sqrt{\frac{2 \times M \times D}{P \times C}}$$

a Draw a UML class diagram to describe the above.

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

c Write a test function as follows:

- Lewis Hamilton is a racing driver, weighing 67kg.
- The NR-14 is racing car with a chassis of mass 582kg, containing a 5-cylinder TurboCharge2000 engine of mass 91kg and power factor 250.
- The TFD-2 is a racing car with a chassis of mass 411kg, containing a 4 cylinder SupermanV3 engine of mass 87kg and power factor 175.
- The time taken for Lewis Hamilton to race the NR-14 over 1000m is calculated and the results printed out.
- The time taken for Lewis Hamilton to race the TFD-2 over 1500m is calculated and the results printed out.
- The Formula One regulations are changed to state that the minimum racing mass of a car must now be 550kg, and the results of the above two trials re-calculated and re-printed.

d Write the bodies of the functions from part (b). You may assume that the functions `sqrt`, `log`, and `exp` are available from the standard math library.

The four parts carry equal marks.