IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2016

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

PAPER M1

PROGRAM DESIGN AND LOGIC

Tuesday 3 May 2016, 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)*

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, but not equivalences, unless they are proved by natural deduction, themselves.

1 a Show by natural deduction that $p \leftrightarrow q \vdash p \vee \neg q$:

b Show by any technique you wish, except truth tables that
$p \leftrightarrow q \equiv (p \wedge q) \vee \neg(p \vee q)$

c Show that

$$S1,\ S2,\ S3,\ S4,\ S5 \vdash atDoor \rightarrow (takePhoto \vee callBackup)$$

using only natural deduction and parts (a) and (b), above, if required, where *S1- S5* are propositional logic sentences as follows:

S1	$atDoor \rightarrow (goIn \leftrightarrow safe)$
S2	$goIn \wedge safe \rightarrow takePhoto$
S3	$\neg goIn \wedge \neg safe \wedge dayTime \rightarrow takePhoto$
S4	$\neg goIn \wedge \neg safe \wedge nightTime \rightarrow callBackup$
S5	$nightTime \leftrightarrow \neg\ daytime$

You may use the equivalences $\neg\neg X \equiv X$ and $X \vee Y \equiv Y \vee X$ without proof.
For ease of reading and constructing the proof consider proving and using general lemmas for intermediate proofs.

d Show that

$$T1,\ T2,\ T3 \vdash \forall Z\,(p(Z) \rightarrow \exists Y\, t(Z,Y))$$

using only natural deduction and parts (a) and (b), above, if required, where *T1-T3* are predicate logic sentences as follows:

T1	$\forall X\,(\exists Y q(X,Y) \vee r(X) \leftrightarrow p(X))$
T2	$\forall X\ \forall Y\,(q(X,Y) \rightarrow t(X,Y))$
T3	$\forall X\,(r(X) \vee m(X) \rightarrow t(X,X))$

Parts a, b, c, d carry 15%, 15%, 40%, 30% of the marks, respectively.

2 a The following sentences $S1$ - $S5$ can be divided up into two groups such that the sentences in each group are equivalent with one another.

$S1$	$\forall X\ \forall D\ (manager(X,D) \wedge highSales(D) \rightarrow promoted(X))$
$S2$	$\forall X\ \forall D\ (manager(X,D) \wedge \neg highSales(D) \rightarrow \neg promoted(X))$
$S3$	$\forall X\ \neg \exists D\ (manager(X,D) \wedge highSales(D) \wedge \neg promoted(X))$
$S4$	$\forall X\ \forall D\ (manager(X,D) \rightarrow \neg highSales(D) \vee promoted(X))$
$S5$	$\forall X\ \forall D\ (manager(X,D) \rightarrow \neg(\neg highSales(D) \wedge promoted(X)))$

i) Identify the two groups.

ii) Assuming that *manager(X,D)*, *highSales(D)* and *promoted(X)*, respectively, mean "*X is manager of department D*", "*department D has high sales*", and "*X is promoted*", which of the two groups expresses the following:

"*Sales department managers whose departments have high sales are promoted*"?

iii) For any two of the equivalent sentences amongst $S1$ - $S5$ show that they are equivalent, using any techniques you wish.

b Formalise in predicate logic the sentences (i)-(vi), below, that concern a sales department. Use only the predicates listed below and $<, \leq, >, \geq$ for comparison of times and quantities. Ensure that you present your formulas clearly, using brackets to correctly identify the scope of quantifiers and disambiguate where necessary.

order(ID,C,I,Q,T)	*to mean customer C orders quantity Q of item I at time T. ID is the identifier of the order.*
process(ID,C,I,Q,T)	*to mean the order with identifier ID from customer C for quantity Q of item I is processed at time T.*
apology(ID,C,T)	*to mean at time T an apology is sent to customer C for the order with identifier ID.*
invoice(ID,C,T)	*to mean at time T an invoice is issued to customer C for the order with identifier ID.*
dispatch(ID,C,T)	*to mean at time T the order with identifier ID is dispatched to customer C.*
instock(I,Q,T)	*to mean at time T, Q amount of item I is in stock.*
owes(C,T)	*to mean customer C owes money at time T.*
paid(ID,C,T)	*to mean at time T customer C paid the invoice for order with identifier ID.*
sells(D,I)	*to mean department D sells item I.*
wellman(D)	*to mean department D is considered well managed.*

i) If a customer orders a quantity of an item, then at a time after the time of the order either the order is processed or an apology is sent to the customer.

ii) An order is processed at a time if and only if at that time an invoice is issued to the customer for the order and the order is dispatched to the customer.

iii) An apology is never (i.e. at no time) sent for an order to a customer if at the time of the order sufficient stock is available for the quantity of the order.

iv) No order made by a customer is dispatched to the customer at a time he/she owes money.

v) A customer owes money at a time T if he has been issued an invoice at an earlier time and has not paid the invoice since the time of the invoice up to and including time T.

vi) A (sales) department is considered well managed if and only if at all times it has in stock all the items that it sells (i.e. stock quantities of the items are greater than 0).

Parts a, b carry 25%, 75% of the marks, respectively.

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

3 Consider a maritime radio messaging system which enables ships to broadcast "Mayday" messages to nearby ships in the event of an emergency:

- Ships have a call sign (a unique identifying string), a radio broadcast range in miles and a position (lat, lon) where lat and lon are floating point numbers indicating latitude and longitude coordinates in decimal degrees.
- Ships can update their positions.
- In (and only in) an emergency, a ship can broadcast a "Mayday" message reporting its call sign and position. The message will be received by ships within the radio broadcast range. The distance in miles between two positions (lat_1, lon_1) and (lat_2, lon_2) can be computed by the Haversine formula:

$$2r \arcsin\left(\sqrt{\text{hav}(\text{rad}(\Delta lat) + \cos(\text{rad}(lat_1))\cos(\text{rad}(lat_2))\text{hav}(\text{rad}(\Delta lon))}\right)$$

where $\text{hav}(\theta) = \sin^2(\frac{\theta}{2})$, $\text{rad}(\alpha) = \frac{\pi\alpha}{180}$, $\Delta lat = lat_2 - lat_1$,
$\Delta lon = lon_2 - lon_1$, and $r = 3956$ miles (the approximate radius of the earth).

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:

- Ship *Endeavour* with call sign 2FBA7 and a radio broadcast range of 20 miles is at position (36.158, -5.357). Ship *Sea Swan* with call sign 2CEU8 and a radio broadcast range of 10 miles is at position (36.180, -5.390).
- *Sea Swan* experiences an emergency.
- *Sea Swan* broadcasts a Mayday message.
- *Endeavour* updates its position to (36.179, -5.391).

c Write function bodies for your classes (excluding the template class).
Hint: The C++ math library function for arcsin is `asin(...)`.

The three parts carry, respectively, 35%, 25%, and 40% of the marks.

4 Consider the description of the following scenario:

- Human resources (HR) personnel, who assist with applications, have a name, year of birth, and have one letter tray they own, which can hold a number of application forms (AF). There are two types of AFs: a form for networking posts, NAF, and a form for programming posts, PAF. Both forms contain the applicant's name, year of birth, and their module mark (in percent) for Operating Systems. In addition, the NAF contains the module mark for Computer Networks and the PAF contains the module mark for Programming.

- When an HR person receives a new application, they put it on their letter tray if it is not full. Newly received applications are always put on top of already received ones. Occasionally HR personnel can be asked to count the number of applications on the tray or to remove from the tray any applications that do not meet the company-wide minimum application score of 66. This score can be calculated for every AF as shown in the table below. You may assume the availability of the following template class:

```
template <typename T> class stack {
 public:
  stack(); //FIFO container representing a stack
  void push(const T& item); //add item onto the stack
  T& top(); //returns reference to item on top of stack
  void pop(); //removes item from top of stack
  int size(); //returns number of items };
```

Type of Application	Score
Networking (NAF)	(operating_system_mark + networking_mark)/2
Programming (PAF)	(operating_system_mark + programming_mark)/2

a Draw a UML class diagram to describe the above.

b Write C++ class declarations to support the above.

c Write a test function as follows:

- HR person Helen, who was born in 1975, has a tray that can hold up to 50 applications.
- She receives a NAF application from Peter, born in 1986, with a mark of 65 in Operating Systems and 75 in Networks. Later, she receives a PAF as well as a NAF application from Susan, born in 1989, with a mark of 70 in Operating Systems, 60 in Networks, 80 in Programming.
- Helen is asked to remove AFs that do not meet the minimum score.
- Finally, Helen is asked to report the number of AFs that are left.

d Write the bodies of the functions from part (b). You might want to use a temporary stack when applications need to be removed from the tray.

The four parts carry, respectively, 25%, 35%, 15%, and 25% of the marks.