IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2015

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

PAPER M1

PROGRAM DESIGN AND LOGIC

Monday 27 April 2015, 14:00
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 Consider the following sentence:

$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \rightarrow (q \vee s)$

Prove that it is a tautology, in two different ways:

i) By using equivalences only.
ii) By using natural deduction only.

You should not use truth tables in this question.

b For each of the following pairs of sentences state whether or not they are equivalent. If they are equivalent prove it using any techniques you wish. If they are not equivalent then:

- provide an example that illustrates they are not equivalent,
- state if one sentence in the pair entails the other, and if so identify which entails which, and
- prove the entailment using natural deduction only.

i)	$\forall X\,(p(X) \rightarrow q(X))$,	$\forall X\, p(X) \rightarrow \forall X\, q(X)$
ii)	$\exists X\,(p(X) \wedge q(X))$,	$\exists X\, p(X) \wedge \exists X\, q(X)$
iii)	$\forall X\,(p(X) \rightarrow (\exists Y\, q(X, Y) \rightarrow r(X)))$,	$\forall X\,(p(X) \wedge \neg\, r(X) \rightarrow \forall Y\, \neg q(X, Y))$

c Show

$\forall X\,(p(X) \rightarrow q(X)),\ \forall X\,(\neg r(X) \rightarrow p(X)) \vdash \exists Y\,(q(Y) \vee r(Y))$,

using only natural deduction and the knowledge from part(a), above, if required.

Hint: Recall that in natural deduction, you can freely add tautologies to your premises.

Parts a, b, c, carry 35%, 45%, 20% of the marks, respectively.

2 a Formalise in predicate logic the sentences (i)-(iv), below, that talk about tube (underground train) lines and stations. Use only the predicates listed below and "=", if needed. Ensure that you present your formulas clearly, using brackets to correctly identify the scope of quantifiers and disambiguate where necessary.

tubeline(X)	*to mean X is a tube line.*
onNet(X)	*to mean X is a station on the network.*
servedBy(S, L)	*to mean station S is served by line L.*
posChange(L1, L2, S)	*to mean it is possible to change from tube line L1 to tube line L2 at station S.*
travel(P, L)	*to mean P travels on tube line L.*
possess(X, Y)	*to mean X possesses Y.*
validTravelDoc(V, P)	*to mean V is a valid travel document for person P.*
over65(P)	*to mean P is over 65 years old.*
freedomPass(V), dayTicket(V), seasonTicket(V)	*to mean V is a freedom pass, a day ticket, and a season ticket, respectively.*

i) There are two different tube lines (only), and every station on the network is served by at least one of them. Some stations on the network are served by one tube line only.

ii) Each tube line serves a number of stations (more than one station) on the network. It is possible to change from one tube line to a different one at a station if and only if both lines serve that station.

iii) One does not travel on the tube unless one has a valid travel document.

iv) The valid travel documents are the following: a freedom pass, if the passenger is over 65 years old, a day ticket, and a season ticket.

b The tube managers would like to introduce a new additional ticketing system, called *single ticket*, that charges passengers for travelling from one station to another, according to a tariff. Suppose the tariff is specified by the predicate *cost(S1, S2, C)*, specifying that the cost of single (i.e. one way) travel from station *S1* to station *S2* is *C*. Using the predicates *cost*, *hub(S)* to mean *S* is a hub station (i.e. is on more than one tube line), and any comparison predicates, =, <, >, formalise the following in predicate logic. Note that you do not need to define *hub(S)*, you can simply use it.

i) With single tickets the cost of travelling from one station to another is the same regardless of the direction, i.e. the cost of travelling from S1 to S2 is the same as the cost of travelling from S2 to S1.

ii) The cost of a single ticket starting from a hub station is greater than a single ticket starting from a non-hub station.

Suppose *a* is a hub station and *b* is a non-hub station and *cost(b, a, 10)*. Show, using any techniques you wish, that this information together with sentences b (i) and (ii) is inconsistent. You can freely use properties of the comparison predicates, such as

$\neg \exists X, Y\ (X=Y \wedge X<Y)$, or even more simply $\forall X \neg(X<X)$, or $\forall X \neg(X>X)$.

Parts a, b carry 70%, 30% of the marks, respectively.

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

3 Consider a security system for protecting residential properties:

- Several *security guards* may be assigned to protect a *property*.
- *Motion detectors* can be installed to monitor particular areas within a property. Each motion detector is associated with a *location description* that describes the area monitored by that motion detector.
- A motion detector *activates* when it detects movement. When this happens, an alert message including the location description of the motion detector is sent to all security guards assigned to protect the corresponding property.

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 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:

- Kensington Palace is a property fitted with three motion detectors having location descriptions "Hallway East", "Hallway West" and "Crown Jewels Display Case" respectively.
- Imperial College is a property fitted with a single motion detector having location description "Rector's Office".
- Alice and Bob are security guards who are assigned to protect Kensington Palace.
- The "Hallway East" motion detector activates.
- Alice is assigned to protect Imperial College.
- The "Crown Jewels Display Case" motion detector activates.

d Write the bodies of the functions from part (b).

The four parts carry equal marks.

4 Consider the description of the following scenario:

- Employees have a *name* and can have one *laptop*. A new laptop can be ordered at any time, which replaces any existing one. A laptop has a *base cost* and a number of *components*. The components come in two types: *computer memory (RAM)* and *I/O devices*. RAM has a price (in £), make, and size (in GB). I/O devices have a price, make, and response time (in ms).
- Employees can order a new laptop by specifying the base cost and a list of desired components. The list of components has no particular order or size. The order is only successful if the total cost of the laptop is below £850, which is the budget of every employee. Also, an employee can add additional components to their existing laptop as long as the total cost is still within the budget. You may assume the availability of the following template class:

```
template <class T> class vector {
 public:
  vector(); //container representing dynamic arrays
  void push_back(const T& item); //add item at the end
  void pop_back(); //delete last item
  int size(); //returns number of items
  T& operator[]; //returns the item at a given index }
```

- For every laptop an overall performance score can be calculated, which is the sum of the performance scores of the installed components. The performance score of individual components is calculated as shown in the table below.

Component	Performance Score
I/O Device	price $*$ 0.2 + (50/response_time)
RAM	size $*$ 8 + price $*$ 0.1

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:

- Employee Andrew orders a laptop with the base cost of £500 and the following components: 2 memory components that are both from Crucial with the size of 8 GB that cost £90 each, and an I/O device from Samsung with a response time of 5.5ms that costs £150.
- Andrew obtains the overall performance score of his new laptop.
- Andrew adds more memory to the laptop. The component is from Corsair with the size of 4 GB and comes at a price of £50. He obtains the new overall performance score.

d Write the bodies of the functions from part (b). You may assume that a laptop without any installed components has a performance score of 100.

The four parts carry, respectively, 20%, 30%, 25%, and 25% of the marks.