Answer FOUR questions.

If you attempt to answer more than the required number of questions, the mark awarded for the excess question (i.e. the one with the lowest mark) will be discarded.

1 (a). It is a commonly held belief in some sections of the computing community that formal specification notations increase software development costs and, what is more, are incomprehensible to clients. What arguments may be put forward to counter this resistance to the use of formal specifications?
(6 marks)

(b). Why have special purpose specification notations been developed and what technical benefits can result from their use?
(6 marks)

(c). Explain the difference between the model-based and the property-based styles of specification. Which category does the VDM specification notation belong to and why is it particularly well suited to software verification? (6 marks)

(d). Outline the principle of 'design by contract' for software components and explain its relationship to the VDM style of specification. How might such an approach have averted the software fault that led to the explosion of the European Space Agency's Ariane 5 rocket in 1996?
(7 marks)

_____ n

2 (a). Explain what is meant by each of the following terms in the context of the VDM specification language, giving a short illustrative example in each case:
(i). 'map A to B' for given sets A and B;
(5 marks)
(ii). 'seq of C' for some given base set C.
(5 marks)

(b). Consider the following simplified model in the VDM notation for a customer 'loyalty' scheme for frequent flyers with an airline company. Customers who are registered under the scheme are identified by a unique customer number, whose type is Cnum, and are credited with points, termed 'air-miles', each time they take a flight with the airline. Customer details are stored in a composite type CustomerRecord, which has fields to record the customer's name, the customer's current air-miles total of points and the sequence of flight numbers corresponding to the flights which the customer has taken with the airline, i.e.

```
CustomerRecord :: name      : NameType
                  air-miles : N
                  flights   : seq of FlightNum
```

Question continues

2 (b). continued
The state of the system is to be modelled by two maps:

```
customers : map Cnum to CustomerRecord
value     : map FlightNum to N
```

the first of which relates customer numbers to customer details, and the second of which records the fixed air-miles points value of each flight number. Note that the types Cnum, NameType and FlightNum need not be further defined here; the type N refers to the set of non-negative whole numbers.

(i). Give a line-by-line explanation of the following operation specification in VDM and describe its overall effect. (4 marks)

(ii). Provide an implicit VDM specification for the following operation:

```
HOW-MANY-AIR-MILES( c : Cnum ) p : N
```

This operation should return a result p which is the number of air-miles points currently credited to the customer with number c. (3 marks)

(iii). Provide an implicit VDM specification for the following operation:

```
UPDATE( c : Cnum, f : FlightNum )
```

This operation should update the recorded air-miles and flight history of customer with number c, provided the customer has already registered with the scheme. The update uses the fact that the customer has taken flight number f which should be added as the last in the customer's sequence of flights.
(4 marks)

(iv). Provide an implicit VDM specification for the following operation:

```
MOST-FREQUENT-FLYERS( ) r : set of Cnum
```

This operation should return as its result r, the set of all customer numbers whose total of flights taken with the airline is greatest amongst all registered customers. (4 marks)
n

3 (a). (i). Describe the overall syntactic structure of an OBJ algebraic specification (in the ObjEx dialect) and the operational semantics of term rewriting that enable appropriate expressions to be evaluated.
(8 marks)

(ii). Describe, in general terms, the IMAGE feature of ObjEx specifications and explain how it facilitates specification re-use.
(4 marks)

(b). The following OBJ specification is a simplified (and incomplete) system for recording a list of theatrical performances together with seat availability. Each individual performance is denoted as P(p,d,m,s) where p (of type play, a user-defined type), is the title of the performance, d and m (both of type nat, the built-in natural number type) are day and month respectively, representing the date of the performance, and s (also of type nat) is the number of vacant seats left for the performance.

```
Theatre
OBJ Plays
SORTS play
OPS Hamlet, Macbeth, Romeo&Juliet, AsYouLikeIt -> play
JBO

OBJ Performances / Plays
SORTS performance plist
OPS
  *** P(play, day, month, seatsleft) represents a single   ***
  *** performance. The SORT plist is a performance list.   ***
  P     : play nat nat nat  -> performance
  nil   :                   -> plist  *** The empty list   ***
  _ & _ : performance plist -> plist  *** Add performance  ***
  isIn? : play plist        -> BOOL   *** Is play in list? ***
VARS
  p?, p : play
  d?, m?, s?, d, m, s : nat
  rest  : plist
EQNS
  ( isIn?(p?, nil)               = F )
  ( isIn?(p?, P(p,d,m,s) & rest) = T  IF p? == p )
  ( isIn?(p?, P(p,d,m,s) & rest) = isIn?(p?, rest)
                                          IF not p? == p )
JBO

OBJ Test / Performances
OPS ShakespeareTheatre : -> plist
EQNS
  ( ShakespeareTheatre =
      P(Hamlet,16,5,60) & P(Macbeth,17,5,45) &
      P(Romeo&Juliet,18,5,0) & P(Hamlet,19,5,40) & nil )
JBO.
```

(i). Outline the steps in the term rewriting evaluation of the following expression:

```
isIn?(Macbeth, ShakespeareTheatre)
```

(3 marks)

Question continues

3 (b). continued

(ii). Give appropriate equations to incorporate the following operation:

count : play plist -> nat

This operation should deliver a total count of all vacant seats summed over all performances of a particular play (the first parameter) in a given performance list (the second parameter), e.g.

count(Hamlet, ShakespeareTheatre)

should give 100 as its result. (4 marks)

(iii). Give appropriate equations to incorporate the following additional operation:

reserve : nat play nat nat plist -> plist

This operation should reserve the required number of seats (the first parameter) for the performance of the given play (the second parameter) on the specified day and month (the third and fourth parameters respectively) in a given performance list (the fifth parameter). The reservation should be put into effect by subtracting the required number of seats from the number of vacant seats left for the performance, providing the result would be non-negative, and all other details in the performance list are to remain unaltered. If there are not enough seats left for the chosen performance, or the performance requested is not present in the list, the entire performance list is to be returned unaltered by this operation. As an example,

reserve(4, Hamlet, 19, 5, ShakespeareTheatre)

should give as its result:

```
P(Hamlet,       16, 5, 60) &
P(Macbeth,      17, 5, 45) &
P(Romeo&Juliet, 18, 5,  0) &
P(Hamlet,       19, 5, 36) & nil
```

(6 marks)

n

4 (a). (i). Describe briefly the components of a Z schema and state the conditions under which conjunction and disjunction of schemas may take place. (4 marks)

A specification in Z for the membership system of a tennis club consisting of at most 50 people, is to be based on the state schema given below, which records the name and telephone number of each member. Note that NAME and NUMBER are basic sets which need not be further specified here. In what follows you may also assume the existence of the given set:

REPORT = {success, alreadymember, novacancy}

(ii). Give a Z schema for an operation Join which adds a new name and associated telephone number to the club, provided that there is a vacancy. The Join schema should also report back the output result success. (4 marks)

(iii). Explain the notion of 'variable hiding' and state how this may be used to obtain the precondition of a Z schema. Illustrate your answer by deriving the schema PreJoin representing the precondition of the Join schema. (4 marks)

(iv). Assuming the existence of appropriate error-reporting schemas AlreadyMember and NoVacancy which report back the output results alreadymember and novacancy respectively, use the schema calculus to formulate RJoin, a 'robust' version of the Join schema. (1 mark)

(b). (i). Explain the pictorial syntax of marked Petri nets and the semantic rules that enable a marked Petri net to be animated. (5 marks)

With the help of separate small illustrative examples, show how each of the following can be modelled with marked Petri nets:

(ii). synchronisation of activities; (2 marks)

(iii). non-shareability of some resource. (4 marks)

(iv). Explain in words how your last example could be modified if the resource, instead of being non-shareable, was in fact shareable by two processes simultaneously. (1 mark)

n

5 (a). Explain, in general, why formal verification cannot provide an absolute guarantee of perfect software.
(5 marks)

(b). State, with appropriate explanations, Hoare's proof rules for:
(i). assignment to a simple (non-array) variable;
(3 marks)
(ii). partial correctness of a while loop.
(3 marks)

(c). Consider the following Ada function which computes the sum of squares of the first n natural numbers, provided n is greater than zero. An incomplete proof tableau is provided.

(i). What is the loop invariant? (2 marks)

(ii). Complete the proof tableau by deriving the missing assertions, namely C and F, and by justifying the step from Assertion D to Assertion E. Hence prove that the program is partially correct. (6 marks)

(iii). Identify a suitable loop variant function and hence prove total correctness.
(4 marks)
(iv). Despite the fact that the sum_of_squares function can be proved totally correct, under what circumstances might problems ensue when it is run on a real machine. (2 marks)
n