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). By referring to the American Airlines seat reservation system of the mid-1960s, i.e. the SABRE system, outline the symptoms of the so-called 'software crisis' that led to the emergence of software engineering as a discipline. (4 marks)

(b). With the help of a diagram describe the classic 'waterfall' model of software development and list its merits and drawbacks. (8 marks)

(c). Briefly explain what is meant by a software process improvement model and list, together with brief descriptions, the five levels in the Capability Maturity Model (CMM). (8 marks)

(d). By referring to the emergency call handling system of the London Ambulance Service that was operational for a short while in the early 1990s, discuss some of the simple lessons that can be learnt from such attempts to build large software systems, even now, thirty years after the advent of software engineering. (5 marks)

n

2 (a). Give a checklist, together with brief descriptions, of the potential problems or pitfalls for developers of natural language requirements documents, i.e. the so-called 'seven sins of the specifier'.
(7 marks)

Identify any such problems in the following extract from a natural language requirements document for a mail-order shopping system.
(4 marks)

The Bigwood mail-order company will only accept customer orders by surface mail on standard forms issued with the home shopping catalogue. All orders must be accompanied by payment in full. The company does not have the staff to handle telephone orders, although it will accept faxed orders if they too are on the standard forms and are accompanied by credit card payment details. The company realises that it urgently needs to consider accepting orders from the Internet, if it is to retain its share of the home shopping market. When orders are received, a product file is queried to determine whether there is sufficient level of stock to satisfy the order. If the stock level of any item is insufficient, or merely deemed to be rather low, further supplies are to be reordered from the supplier. When an order has been priced and validated, it is assigned a unique fifteen character order code and stored in an order file. After payment has been cleared, a payment confirmation flag is set and the actual goods are sent to the customer. At any time the manager can request a report of orders made, but the default situation is to generate such a report automatically every few weeks.

(b). The accompanying diagram is intended to be a data flow diagram (DFD) illustrating the requirements of the simplified mail-order system as described in part (a). The DFD contains a number of errors and other undesirable aspects which should be listed and explained.
(6 marks)

Use the given diagram to produce a context diagram and a first level decomposition diagram, both of which should be free of the problems you have just listed. Note that you should not extend the functionality of the system.
(8 marks)

n

3 (a). OBJ is a language for writing algebraic specifications, by means of equations.

(i). Explain the process of 'term-rewriting' by which expressions may be evaluated in OBJ. (2 marks)

The following is an incomplete OBJ specification of a simplified system to monitor the level of some liquid chemical in a storage tank. The operator empty creates an empty tank. The operator add enables a given quantity of the chemical, specified as a natural number, to be added to the tank. The operator use enables a given quantity to be taken from the tank. Operator level? returns the quantity of liquid in the tank, max is the maximum capacity of the tank and full? returns 'true' if the tank is full to capacity and 'false' otherwise.

```
MonitorTank
OBJ Tank
SORTS tank
OPS
   empty  :           -> tank  *** create empty tank ***
   add    : nat tank  -> tank  *** add to tank       ***
   use    : nat tank  -> tank  *** use from tank     ***
   level? : tank      -> nat   *** how much in tank? ***
   max    :           -> nat   *** tank capacity     ***
   full?  : tank      -> BOOL  *** is tank full?     ***
VARS
   t : tank
   q : nat
EQNS
   ( max          =  50                  )
   ( full?( t )  =  level?( t ) == max )
   ***  SOME EQUATIONS MISSING FROM HERE  ***
JBO.
```

Give the missing equations appropriate for specifying each of the following:

(ii). the level of the empty tank;
(1 mark)

(iii). the level after adding quantity q;
(3 marks)

(iv). the level after using quantity q.
(3 marks)

Note that it is not permitted to fill the tank beyond its maximum capacity, nor to use more chemical than is in the tank.

(v). Illustrate the 'term-rewriting' process by giving the steps in the evaluation of the following expression:
(4 marks)

full?(use(30,add(50,empty)))

Question continues

3 (b). In the context of the VDM approach to formal software specification, explain what is meant by:

(i). a pre-condition; (1 mark)
(ii). a post-condition; (1 mark)
(iii). a data state invariant.
(1 mark)

Consider a simplified model in VDM of an estate agent's information system, recording addresses of properties and whether they are currently for sale, under offer or sold. An appropriate abstract type Address is assumed and need not be further defined here. The state of the system consists of three components: for-sale, which is the set of addresses currently registered with the estate agent as available for purchase; under-offer, which is the set of addresses for which an offer of purchase has been made; and sold, which is the set of addresses for which a sale has been completed. It can be assumed that any given address is in, at most, one of the three sets at any point in time. In addition, all sets can be considered to have been initialised as empty.

```
for-sale       : set of Address
under-offer : set of Address
sold           : set of Address
```

(iv). Provide the data state invariant for this situation.
(2 marks)

Construct implicit specifications in the VDM notation for the following two operations:

(v). MAKE-OFFER(a : Address)

Provided address a is currently registered as available for purchase, this operation changes the state, recording a as now under offer.
(3 marks)

(vi). ENQUIRE (a : Address) r : Status

This operation delivers as its result r, reporting the appropriate status of address a. The type Status is defined as:

```
Status = { FOR-SALE, UNDER-OFFER, SOLD,
           ADDRESS-NOT-KNOWN }.
```

(4 marks)

n

4 (a). List the possible types of module cohesion giving a brief description of each and show how they form a qualitative cohesion spectrum.
(8 marks)

State, with reasons, the type of cohesion which best describes each of the following two modules given in outline:
(4 marks)

(i). Module SHUT-DOWN-PLANT()
```
Turn off heating supply;
Close input valves;
Send alert to chief operator's console;
Send diagnostics to fault log;
```

(ii). Module CREDIT-CARD-SERVICES(in CUSTOMER-ID, in FLAG)
```
Retrieve customer details using CUSTOMER-ID;
case
  when FLAG=1: change credit limit;
  when FLAG=2: cancel card;
  when FLAG=3: send new card;
end case;
```

(b). Outline the underlying rationale and principles of the Jackson Structured Programming (JSP) approach to software design.
(2 marks)

Explain the term 'structure clash' in the context of JSP.
(2 marks)

A computer systems manager decides to monitor the time spent at terminals by staff in the organisation. The LOGIN and LOGOUT commands are modified, so that information in the following order:

DATE TIME TERMINAL-ID USER-ID COMMAND

is appended as a single monitoring record at the end of a system file whenever a command of type LOGIN or LOGOUT is performed. This file is to be used periodically as input to a system utility which will produce as output a list of terminal session records in chronological starting order, i.e. ordered by LOGIN date and time. A terminal session is the period a user spends logged-in at a terminal and a terminal session record should contain the following sequence of information:

USER-ID LOGIN-DATE LOGIN-TIME TERMINAL-ID SESSION-DURATION

Draw JSP diagrams to represent the input and output structures for this situation.
(7 marks)

Explain why these give rise to a structure clash and, in general terms, suggest a simple way of overcoming this problem.
(2 marks)

n

5 (a). In the context of software testing, explain what is meant by the following terms:

(i). a test frame in the category-partition approach;
(2 marks)

(ii). component coverage;
(1 mark)

(iii). an equivalent mutant.
(1 mark)

(b). A program P takes two integers a and b as input. A test specification in the Test Specification Language (TSL) is initially developed as follows.

```
VALUE-OF-a:
      below-zero
      equals-zero
      above-zero
VALUE-OF-b:
      below-zero
      equals-zero
      above-zero
COMPARING:
      a-equals-b
      a-not-equals-b
```

(i). Enumerate all test frames and determine which ones are valid.
(9 marks)

(ii). Hence determine a succinct characterisation of the valid combinations. Your characterisation need not be in the TSL notation.
(2 marks)

(c). Consider the following implementation of program P, given in pseudocode.

```
1    INPUT a, b;
2    x = 0;  y = 0;
3    IF ( a >= 0 )
4    THEN x = (x+1)*a;
5    ENDIF;
6    IF ( NOT ( b == a ))
7    THEN y = (y-1)*b;
8    ENDIF;
9    OUTPUT x, y;
```

(i). Draw a flowgraph of program P.
(2 marks)

Hence list a minimal set of paths to achieve each of the following:

(ii). complete coverage of all nodes, i.e. TER1=1;
(1 mark)

(iii). complete coverage of all edges, i.e. TER2=1;
(2 marks)

(iv). complete coverage of all paths.
(2 marks)

(v). Explain why complete path coverage is not normally possible for most programs. (1 mark)

(vi). What can be deduced about the mutant version of P formed by replacing the condition IF (a >= 0) on line 3 by IF (a > 0)? Explain your reasoning.
(2 marks)

n