THE UNIVERSITY
of LIVERPOOL

JANUARY 2003 EXAMINATIONS

Master of Science : Year 1

SOFTWARE ENGINEERING

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer **FOUR** questions

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

THE UNIVERSITY
of LIVERPOOL

1(a). Describe what is meant by the attributes of 'robustness' and 'maintainability' for quality software and outline, in general terms, what the software developer can do to optimize these characteristics. **(6 marks)**

(b). With the help of a diagram, briefly outline the essential features of Boehm's spiral model of the software development process as modified by Pressman. Explain how it can accommodate both the classic 'waterfall' model and a more evolutionary model. Note that, whilst you need only indicate the general features of the spiral model, your explanation should include the meaning of each sector in the diagram. **(6 marks)**

(c). State the 'Law of Continuing Change' and the 'Law of Increasing Complexity' that were postulated by Lehman and Belady. Describe how these 'laws' were derived and use them to explain why software which is developed using an evolutionary approach may be more difficult to maintain. **(7 marks)**

(d). Briefly describe the types of maintenance that software may undergo and identify two of the most significant managerial factors that may affect the cost of maintenance.
(6 marks)

∎

THE UNIVERSITY
of LIVERPOOL

2(a). Compare and contrast the use of data flow diagrams (DFDs) and use-case diagrams in the software development process. **(4 marks)**

(b). Draw a use-case diagram to represent the following situation. **(6 marks)**

A diagram drawing tool for a PC enables a user to create a new diagram by opening a drawing window where the current diagram can be modified as required. The user can store the current diagram in a diagram folder and may open a previously stored diagram so that it becomes the current diagram and hence can be modified. Stored diagrams in the diagram folder can be deleted only by a special administrative user who, in every other respect, has the same facilities as an ordinary user.

(c). Using the DFD notation, represent the following description of an online seat reservation system for a theatre by drawing:

(i). a context diagram; **(6 marks)**

(ii). a first-level decomposition diagram. **(9 marks)**

A web-based online seat reservation system is to be developed for a theatre. A customer wishing to make an enquiry provides details of the performance and the number of seats needed. The system responds with information on available seats and their total cost. If the customer wishes to proceed with reservation of the suggested seats, the customer must supply credit card details and then the system transmits all necessary transaction information to the credit card company who respond with a transaction acknowledgement. The customer then receives payment confirmation and seat confirmation. Theatre staff may use the system at any time to request a report detailing unsold seats for each performance.

Internally the system maintains a database of seat bookings for each performance. This database is queried when a customer makes an initial enquiry and it is updated to reflect the customer's reservation of seats once credit card details have been submitted by the customer and a transaction acknowledgement has been received from the credit card company. The database is also queried when theatre staff request a report on unsold seats.

3(a)(i). Distinguish between functional and non-functional requirements of software systems.

(2 marks)

(ii). Consider the following extract from a natural language requirements document and state which type of requirement it is: functional or non-functional. **(1 mark)**

The system should provide a response to queries within a reasonable time period.

(iii). Describe what, if anything, is wrong with the above requirements extract and suggest how it might be improved. **(4 marks)**

(iv). Distinguish between requirements validation and requirements verification and then suggest what approaches are available to the analyst for both activities. **(4 marks)**

(b). Below is a Z schema that specifies the state space of a simplified system for recording the status of each video belonging to a video rental shop.

```
┌─────── VideoRentalShop ───────
│ stock  :  IP VIDEO
│ status :  VIDEO ↦ LOCATION
│
├───────────────────────────────
│ dom status = stock
└
```

The abstract type VIDEO can be assumed to provide a unique identifier for each physical video and need not be considered in any further detail here. In addition, LOCATION can be regarded as the given set:

$$LOCATION = \{inShop, onLoan, Lost\}$$

(i). Explain the components of the above schema in your own words. **(4 marks)**

(ii). Give a Z schema that specifies an operation HowManyOnLoan which returns as output the number of videos out on loan. **(3 marks)**

(iii). Give a Z schema that specifies an operation Rent which records the fact that a given video has been rented, assuming that previously it was available for rent. **(5 marks)**

(iv). In your Rent schema, identify which predicate (or predicates) are pre-conditions and which predicate (or predicates) are post-conditions. **(2 marks)**

■

THE UNIVERSITY
of LIVERPOOL

4(a). Consider the following scenario.

A personnel file is sorted by grade of employee and each employee record contains details of that employee's qualifications. It is required to produce a report with a suitable heading, followed by a list giving the number of staff in each grade who possess a Bachelor's degree, followed in turn by a grand total of all such staff.

(i). Draw logical data structures in the Jackson Structured Programming (JSP) notation for the input personnel file and the output report file. **(8 marks)**

(ii). Identify the correspondences between the personnel file and the report file that enable the program structure given below to be produced. **(2 marks)**

(iii). Convert the given program structure to its equivalent JSP pseudo-code. **(5 marks)**
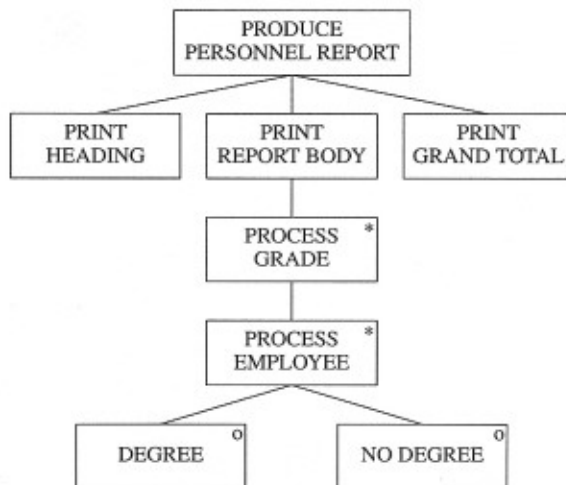
(b)(i). In a state diagram of the Unified Modeling Language (UML) what is meant by a guarded transition and how is it denoted? **(2 marks)**

(ii). Consider the following description of a simplified CD player and then draw a UML state diagram modeling its behaviour. **(8 marks)**

A simple CD player consists of a drawer to hold the CD that is currently being played and an interface consisting of an on/off power switch and three buttons labeled 'load', 'play' and 'stop'. Initially when the power is switched on the drawer is shut. The load button causes the drawer to open if it is currently shut, and to shut if it is open. If there is a CD in the drawer, pressing the play button causes the CD to be played. If it is pressed when the drawer is open, the drawer shuts before the playing starts. If there is no CD in the drawer, pressing the play button causes the drawer to be shut if it is open and to remain shut if it is already shut. The stop button causes the player to stop playing. Button presses other than those explicitly mentioned have no effect, i.e. the CD player remains in whatever its current state is. Finally, power may be switched off no matter what state the player is in. ∎

5(a)(i). Outline the steps in the so-called 'category-partition' method of functional testing.

(6 marks)

(ii). Identify suitable categories and partitions for testing software that implements the following function. You are not required to develop a full test specification in the Test Specification Language (TSL). **(6 marks)**

CONVERT is a function which converts an array, C say, of exactly six characters to its equivalent integer value. The characters to be converted must be decimal digit characters and must be right justified in C. If the input constitutes a negative number, a minus sign occupies the character position immediately to the left of the most significant digit character. If the number of digit characters, together with the possible minus sign, is less than six characters, C is padded with blanks on the left. Because of built-in compiler-checking there is no need to test CONVERT using arrays of more or less than six characters long.

(b). Let *P* be the following pseudo-code program where e1, e2 and e3 represent Boolean expressions and s1, s2, s3 and s4 represent simple statements.

```
1       start;
2       if e1
3       then s1;
4       end if;
5       s2;
6       while e2
7       begin loop
8            if e3
9            then s3;
10           else s4;
11           end if;
12      end loop;
13      stop;
```

(i). Draw the control flow graph for program *P*. **(4 marks)**

(ii). Identify the branches of program *P* and, with the aid of its control flow graph, devise a set of test paths which, if executed, would achieve complete branch coverage of *P*.

(5 marks)

(iii). State a formula for McCabe's 'complexity' measure in terms of the nodes and edges in a control flow graph and then use this formula to calculate McCabe's measure for *P*.

(3 marks)

(iv). Suggest a reason against the use of McCabe's measure as a *general* indicator of program complexity. **(1 mark)**

■