PAPER CODE NO.
COMP509

THE UNIVERSITY
of LIVERPOOL

JANUARY 2005 EXAMINATIONS

Master of Science : Year 1
Master of Science : Year 2

SOFTWARE ENGINEERING

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer **FOUR** questions only

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). With the help of a diagram, outline the classic 'waterfall' model of software development and discuss its strengths and its weaknesses. Name one radical alternative approach to software development. **(10 marks)**

(b). Describe what is meant by the attributes of 'usability' and 'maintainability' for software, and discuss for which group of stakeholders they have most significance in the development of quality software and why. List the other attributes, which together with usability and maintainability, constitute the ISO 9126 standard? **(6 marks)**

(c). Outline the problems that occurred with the software for the emergency call handling system of the London Ambulance Service (LAS) that was operational for a short while in the early 1990s. List the key findings of the enquiry that ensued and discuss what software engineering lessons can be learned. **(9 marks)**

■

2(a). Consider the following description of a simplified video rental system.

With the aid of a member of staff, new customers may register as users of a video rental outlet by supplying their details such as name, address and telephone number. In return, each new customer is given a rental card with a bar code representing their unique identification number. Each video also possesses a bar code representing its unique identification number. When a customer wishes to rent a video, they present their rental card, together with the chosen video and the appropriate payment. A rental receipt is issued for the customer. When a video is returned by a customer, the video's bar code is scanned so that the system's records can be updated as necessary. Members of staff can add videos to, or withdraw videos from, the recorded stock by supplying appropriate details. In addition, staff can produce a management report, on request, in order to identify the most and least popular videos.

Represent this system by drawing:

(i). a use-case diagram in the notation of the Unified Modeling Language (UML); **(5 marks)**

(ii). a context diagram using the notation of data flow diagrams (DFDs). **(5 marks)**

(b)(i). Consider the following more detailed description of the same simplified video rental system and hence draw, using the DFD notation, a first-level decomposition diagram of your DFD context diagram developed in part (a) of this question. **(10 marks)**

Internally the system stores customer details in a customer master file with a field in each customer record to indicate the number of videos the customer currently has out on rental. A video master file is also maintained with a record for each video containing: its identification number, its title, its rental price, its status (in shop, out on rental, or withdrawn) and, if it is being rented, the identification number of the customer currently renting it. A separate rental transaction log keeps details of each video rental with information that includes date and time of rental, customer and video identification numbers. When a new customer registers with the rental outlet, a new customer record is added to the customer master file. When a customer rents a video, the customer and video master files are updated as necessary and a new video rental record is added to the rental transaction log. Payment is handled by a separate process which is also responsible for printing the rental receipt. When a video is returned, its identification number is used to extract its most recent rental record from the rental transaction log which is then used to make the necessary updates to the customer and video master files. When a video is added to, or withdrawn from, the stock available for rent, the necessary changes are made to the video master file. Finally, when a request for a management report is received, the necessary details are extracted from the rental transaction log.

(ii). Give appropriate data dictionary entries to describe the video master file. **(3 marks)**

(iii). What criticisms could be levied against the above detailed requirements, firstly in general terms, and secondly, with specific regard to the stored information? **(2 marks)**

■

3(a). Outline three arguments for, and three against, the use of formal notations for the specification of software systems. **(6 marks)**

(b). Consider the following fragment from an informal specification and describe any problems you can identify, using Meyer's checklist of 'seven sins'. **(5 marks)**

The system should include a separate module for printing a monthly management report containing full sales details for each sales representative only if the current month is the last month of the financial year and the year's sales total is significantly less than the year's sales target or the current month's sales total is less than eighty per cent of the previous month's sales total. Paper copies of the reports are to be retained for three years before undergoing secure disposal.

(c). Below is a schema in the Z formal specification notation, appropriate for modeling the state of a simplified information system for an agency concerned with accommodation rentals. The system records the addresses of properties that are available for rent, together with the number of bedrooms for each property. Property addresses are retained in the system, even when they are no longer available for rent, in case they should become available again in the future. An abstract type `Address` can be assumed to provide a unique representation of each address in the system. It need not be considered in any further detail here.

Accommodation

$$\begin{aligned} forRent &: \mathbb{P}\ Address \\ rented &: \mathbb{P}\ Address \\ bedrooms &: Address \pfun \mathbb{N} \end{aligned}$$

$$forRent \cap rented = \{\}$$
$$dom\ bedrooms = forRent \cup rented$$

(i). Explain the structure and contents of the above schema. **(5 marks)**

(ii). Give a Z schema that specifies an operation `HowManyForRent` which takes as input the number of bedrooms required by a client and outputs the number of properties available for rent that have the required number of bedrooms. **(4 marks)**

(iii). Give a Z schema that specifies an operation `NewAddressForRent` which takes a new address (i.e. one not already recorded in the system) as input, together with its number of bedrooms, and updates the state appropriately. **(5 marks)**

■

4(a)(i). Identify what aspect of an object-oriented design a *state diagram* in the Unified Modeling Language (UML) portrays and explain in general terms how it is realized. **(3 marks)**

(ii). Draw a UML state diagram to model the following scenario. You should take care to label each transition, not just with the event that causes it, but also with a guard and associated actions where appropriate. **(7 marks)**

An automatic ticket vending machine is initially in an 'idle' state. When a customer wishes to use the machine they must first press an appropriate button to select the required ticket type which has an associated cost. This has the effect of taking the machine into a 'paying for ticket' state. The customer then needs to insert a coin worth a known amount, and repeatedly do this as long as the total inserted is less than the cost of the requested ticket. When a coin is inserted such that the total value inserted is sufficient to pay for the ticket, the ticket and any change are issued, and the machine returns to the idle state. If at any stage while paying for a ticket the customer presses a ticket selection button, it has no effect other than displaying a warning message. The message explains that once a ticket type has been selected it cannot be altered but that the customer can press the 'cancel' button which returns all coins and allows the customer to start afresh. Note that inserting a coin while the machine is in the idle state has no effect other than to return the coin. Similarly, pressing the cancel button in the idle state has no effect.

(iii). Identify any two significant limitations or deficiencies in the simplified scenario that would need consideration for a real-world system of this kind. **(2 marks)**

(b)(i). Outline the underlying rationale and principles of the Jackson Structured Programming (JSP) approach to software design and indicate for what type of situation the method is best suited. **(4 marks)**

(ii). Draw a JSP diagram to represent the input for the following scenario. **(7 marks)**

A security checking program is to be developed which scans a computer user's email Inbox looking for messages with executable file attachments (i.e. file type .exe). An Inbox consists of any number of messages and each message consists of a header section, followed by the message body, followed by an attachments section, followed by a signature block. The attachments section consists of a file count followed by zero or more files, each with a three letter file type.

(iii). Identify any two significant limitations or deficiencies in the simplified scenario that would need consideration for a real-world system of this kind. **(2 marks)**

■

5(a). Explain the difference between black-box testing and white-box testing and then state to which of these two types each of category-partition testing and mutation testing belong. **(3 marks)**

(b). A program *P* takes three integers *a*, *b* and *c* as input. It is known from the requirements that the functional behaviour of *P* varies according to the value of input *a* in relation to two fixed boundary values of 0 and 9, and according to the value of input *b* in relation to the single fixed boundary value of 0. It is also known that *P* has the same functional behaviour for all values of input *c*. Devise an appropriate test specification in the Test Specification Language (TSL) suitable for testing *P* using the category-partition approach and, in the absence of any constraints linking *a*, *b* and *c*, determine how many test frames would be generated by your test specification. **(5 marks)**

(c). Consider the following implementation of *P* in pseudo-code.

```
1    INPUT(a, b, c);
2    x = c;
3    if (a > 0 AND b == 0)
4    then x = x + a;
5    end if;
6    if (a > 9 OR b > 0)
7    then x = x + b;
8    end if;
9    OUTPUT(x);
```

(i). Draw a control flow graph for program *P*, with a node for each line, and use the graph to enumerate the full set of distinct paths through *P*. **(6 marks)**

(ii). Identify the path taken for each of the test cases:

$$TC1 \equiv (a = 10,\ b = 0,\ c = 3)$$
$$TC2 \equiv (a = 0,\ \ b = 0,\ c = 3)$$

and hence determine the levels of statement coverage and branch coverage achieved by TC1 alone, and also by the combined effect of TC1 and TC2. **(4 marks)**

(iii). Consider the mutant version of P formed by replacing "`if (a > 0 AND b == 0)`" on line 3 by the faulty statement "`if (a > 0 OR b == 0)`". Determine the necessary constraints on the inputs of *P* that will guarantee exposure of the fault (i.e. will guarantee that the mutant will be 'killed'). Hence, or otherwise, confirm that neither TC1 nor TC2 kills the mutant. Comment on whether your TSL test specification from part (b) would generate a test case to kill the mutant. **(7 marks)**

■