98-Comp-A6

Software Engineering

3 Hours Duration

Notes:

1. If doubt exists as to the interpretation of a question, the candidate is urged to submit with the answer paper a clear statement of any assumptions made.
2. No calculators permitted. This is a closed book exam.
3. Answer any five of the eight questions.
4. Any five questions constitute a complete paper. Only the first five questions as they appear in your answer book will be marked.
5. All questions have equal weight.

Marking Scheme

1. (a) 10 marks; (b) 10 marks.
2. (a) 10 marks; (b) 10 marks.
3. 20 marks.
4. (a) 10 marks; (b) 10 marks.
5. (a) 5 marks; (b) 15 marks.
6. (a) 5 marks; (b) 5 marks; (c) 10 marks.
7. (a) 10 marks; (b) 10 marks.
8. (a) 8 marks; (b) 5 marks; (c) 7 marks.

Total mark out of 100.

Question 1. *The Software Development Process.*

(a) List the stages of the software development life cycle and briefly describe each stage.

(b) Contrast and compare these stages to the stages of purchasing and owning a piece of equipment, such as a car or a refrigerator. In particular, contrast the life cycle costs of owning the equipment with that of software. How are they similar and how are they different? Justify your answer.

Question 2. *Software Design.*

(a) Discuss the differences between object-oriented and function-oriented design.

(b) Sketch a *function-oriented* design of the following system:

> A gas station is to be set up for fully automated operation. A driver inputs his or her credit card into the pump, the card is verified by communication with a credit card company computer and a fuel limit is established. If the card is invalid, it is returned by the pump with no fuel to be dispensed. The driver may then take the fuel required and on completion of delivery (i.e., when either the fuel limit is reached or the pump hose is returned to its holster), the driver's credit card is debited with the cost of the fuel taken..

Question 3. *Object-oriented Design.*

Using an object-oriented approach, derive a high-level design for the system outlined below. Make reasonable assumptions about the system and state them clearly.

> An automatic date-book system for keeping track of daily appointments electronically.

Question 4. *Software Reuse and Portability.*

(a) In an object-oriented programming language, *information-hiding* and *inheritance* can be used to adapt software components for reuse. Describe information-hiding and inheritance, and the pros and cons of using each to support code reuse.

(b) You have been assigned the task of implementing a calendar and clock which gives time and date information. This has to operate on a range of computers from 8-bit micros to 64-bit special purpose processors. Design and implement an abstract data type for representing the calendar and clock that can be readily ported from machine to machine.

Question 5. *Requirements Specification.*

(a) Discuss briefly the problems of using natural language for requirements specification.

(b) Discover ambiguities or omissions in the following statement of requirements for part of a ticket issuing system.

> A ticket issuing system is intended to automate the sale of rail tickets. Users select their destination, and input a credit card and a personal identification number. The rail ticket is issued and the credit card account is charged with its cost. When the user presses the start button, a menu display of potential destinations is activated along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a personal identifier. When the credit card has been validated, the ticket is issued.

Question 6. ***Software Testing.***

(a) Contrast "black-box" testing to "white-box" testing. What are the pros and cons of each approach?

(b) Explain why testing can only detect the presence of errors but not their absence.

(c) Give a set of black-box test cases for the following software components:

1. A sort routine that sorts arrays of integers.
2. A routine that takes a line of text as input and counts the number of non-blank characters in that line.
3. An abstract data type called STRING that provides operations on character strings, including concatenation, length and sub-string selection.

Question 7. *Software Safety.*

A software system is to be developed for a microprocessor-based *Insulin Delivery System* (IDS) in a hospital. The system works by using a micro-sensor embedded in the patient to measure blood parameters that are proportional the sugar level. These parameters are then sent to a pump controller. This controller computes the sugar level, judges how much insulin is required and sends signals to a miniaturized pump to deliver the insulin via a permanently attached needle.

A low blood sugar level, even for a short term, is a serious condition that can result in brain damage and ultimately death. A high blood sugar level, for a long term, can result in eye damage, kidney damage and heart problems.

(a) Conduct a software hazard analysis of the IDS described above. What are the hazards that can occur in the system? What is the risk associated with each hazard?

(b) Using fault tree analysis, discover the conditions that might cause each of the hazards you identified above.

Question 8. *Software Project Management.*

(a) Identify the main stages of risk management in software engineering projects.

(b) Explain why the best programmers do not always make the best software managers.

(c) You are asked by your manager to deliver software to a schedule that you know can only be met by asking your project team to work unpaid overtime. All team members have young children. Discuss whether you should accept this demand from your manager or whether you should persuade your team to give their time to the organization rather than to their families. What factors might be significant in your decision?