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.

Question 1. *The Software Development Process.*

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

(b) In percentage of total effort, how much effort does each stage require on average in industry? Explain your answer.

Question 2. *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 3. *Software Design.*

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

(b) Sketch object-oriented and function-oriented designs of the following system:

> A cruise control system for a car which maintains a constant speed set by the driver. The system should adjust the car controls depending on measured road speed.

Question 4. *Object-Oriented Design.*

Identify possible objects in the following system, and develop an object-oriented design for it. You may make any reasonable assumptions about the system when deriving the design.

> A group calendar and time management system intended to support the timetabling of meetings and appointments across a group of co-workers. When an appointment is to be made that involves a number of people, the system finds a common slot in each of their calendars and arranges the appointment for that time. If no common slots are available, it interacts with the user to re-arrange his or her calendar to make room for the appointment.

Question 5. *Rapid Software Development.*

(a) Explain why the rapid delivery and development of new systems is often more important to businesses than the detailed functionality of these systems.

(b) A charity has asked you to prototype a system that keeps track of all donations that have been received. This system has to maintain the names and addresses of donors, their particular interests, the amount donated, and when the donation was made. If the donation is over a certain amount, the donor may attach conditions to the donation (e.g., the money must be spent on a particular project), and the system must keep track of these and how the donation was spent.

Discuss how you would prototype this system.

Question 6. *Software Testing.*

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

(b) Explain why it is not necessary for a program to be completely free of defects before it is delivered to its customers. To what extent can testing be used to validate that the program is fit for its purpose.

(c) Discuss the differences between functional and structural testing and suggest how they may be used together in the defect testing process.

Question 7. *Critical Software Development.*

(a) Describe the three complementary approaches to developing dependable software.

(b) Describe four software engineering techniques that can lead to fault-free software.

(c) Illustrate how the techniques you describe in part (b) above can be used in the design of a software-controlled insulin delivery system that works by using micro-sensors embedded in the patient to measure some blood parameter that is proportional to sugar level and then control a pump to dispense the necessary amounts of insulin via a permanently attached needle.

Question 8. *Software Validation.*

Explain why sequential software designs are easier to validate than designs that involve parallel processes.

Marking Scheme

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