May 2015

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) 2 marks; (c) 8 marks.
2. (a) 5 marks; (b) 15 marks.
3. 20 marks.
4. (a) 5 marks; (b) 5 marks; (c) 10 marks.
5. (a) 5 marks; (b) 5 marks; (c) 10 marks.
6. (a) 5 marks; (b) 2 marks; (c) 3 marks; (d) 10 marks.
7. 20 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) In percentage of total effort, how much effort does each stage require on average in industry? Explain your answer.

(c) Contrast and compare these stages to the stages of building and owing a house. Comment on how good the analogy is between the software development process and the processes of building and owning the house.

Question 2. *Function Oriented Software Design.*

(a) Explain what is meant by function-oriented software design.

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

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

Question 3. *Object Oriented Software Design.*

A software system is to be developed for a microprocessor-based *Home Security System* (HSS). The system receives input from entry sensors, smoke sensors, temperature sensors and flood sensors. The system is capable of generating alarms, turning on selected lights, and calling owner-specified phone numbers. The system is owner-programmable through a keypad. The owner can set thresholds for the sensors, program phone numbers and set delays for various alarms.

Using an object-oriented approach, derive a design for the HSS described above. Make any reasonable assumption and clearly state them.

Question 4. *Software Testing.*

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

(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) Design a testing strategy for the Home Security System described in **Question 3** above.

Question 5. *Distributed Software Systems.*

(a) Explain why distributed systems are inherently more scalable than centralized systems. What are the likely limits on the scalability of distributed systems?

(b) What is the difference between a fat-client and a thin-client approach to client-server systems development?

(c) Using a distributed object approach, propose an architecture for a national theatre booking system where users can check seat availability and book seats at a group of theatres. The system should support ticket returns so that people may return their tickets for the last-minute resale to other customers.

Question 6. *Real-Time Systems.*

(a) Define real-time software systems.

(b) What is the difference between "soft" real-time systems and "hard" real-time systems?

(c) List 3 examples of computer-based real-time systems. For each example, indicate what "stimuli" feed the system and what devices or situations the system controls or monitors.

(d) Draw a state machine model of the control software for the following real-time system:

> A drink-vending machine that dispenses coffee, with and without milk and sugar. The user deposits a coin and makes his or her selection by pressing a button on the machine. This causes a cup with powdered coffee to be output. The user places this cup under a tap, presses another button and hot water is dispensed.

Question 7. *Software Reliability.*

Select appropriate reliability metrics for specifying the reliability of the system described below. Explain your choices taking into account that some faults are more serious than others. Consider three classes of faults: (1) faults that corrupt data, (2) faults that cause the system to become unavailable, and (3) faults that cause incorrect information to be transmitted to the EPOS terminal. Do you think it is possible to have one reliability metric for all three fault classes?

> A software system controls a network of EPOS (electronic point of sale) terminals in a supermarket. The system accepts bar code information from a terminal, queries a product database, and returns the item name and its price to the terminal for display. The system must be continually available during the supermarket's opening hours.

Question 8. *Software Verification and Validation.*

(a) Discuss the difference between *verification* and *validation*. Explain why validation is a particularly difficult process.

(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) Using your knowledge of Java, C++, C, or some other programming language, derive a checklist of common errors (not syntax errors) that could not be detected by a compiler but that might be detected by program inspection.