National Examinations December 2019

17-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. 20 marks: (a) 10 marks; (b) 5 marks; (c) 5 marks.
2. 20 marks: (a) 6 marks; (b) 7 marks; (c) 7 marks.
3. 20 marks: (a) 10 marks; (b) 10 marks.
4. 20 marks: (a) 5 marks; (b) 5 marks; (c) 5 marks; (d) 5 marks.
5. 20 marks: (a) 5 marks; (b) 5 marks; (c) 10 marks.
6. 20 marks: (a) 5 marks; (b) 5 marks; (c) 10 marks.
7. 20 marks: (a) 5 marks; (b) 5 marks; (c) 5 marks; (d) 5 marks.
8. 20 marks: (a) 10 marks; (b) 10 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. *Software Design.*

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

(b) Using a function-oriented approach, derive a design for an automatic date-book system for keeping track of daily appointments electronically. Make reasonable assumptions about the system and state them clearly.

(c) Using an object-oriented approach, derive a design for the same automatic date-book system described in part (b) above. Again, make reasonable assumptions about the system and state them clearly.

Question 3. *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 home security system.

> 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 generates 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.

Question 4. *Software Testing.*

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

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

(c) Give arguments for and against testing by developers of their own code, as opposed having all testing be the responsibility of a separate team.

(d) 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.

Question 5. *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 6. *Real-Time Software Systems.*

(a) Define real-time software systems.

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

(c) Draw a state machine model of the control software for the following system.

A telephone answering machine that records incoming messages and displays the number of accepted messages on a LED display. The system should allow the telephone owner to dial in, type a sequence of numbers (identified as tones) and have the recorded messages replayed over the phone.

Question 7. *Formal Methods.*

(a) Explain what is meant by "formal methods" in Software Engineering.

(b) What are the advantages of formal methods to software engineers?

(c) What is the difference between algebraic and model-based approaches to formal specification?

(d) An abstract data type representing a stack has the following operations associated with it:

New:	Bring a stack into existence
Push:	Add an element to the top of the stack
Top:	Evaluate the element at the top of the stack
Retract:	Remove the top element from the stack and return the modified stack
Empty:	True if there are no elements on the stack

Define this abstract data type using an algebraic specification.

Question 8. *Distributed Software Systems.*

(a) Explain why deploying software as a service can reduce the IT support costs for a company. What additional costs might arise if this deployment model is used?

(b) A company wishes to move from using desktop applications to accessing the same functionality remotely as services. Identify three risks that might arise and suggest how these risks may be reduced.