National Exams December 2018

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. (a) 10 marks; (b) 5 marks; (c) 5 marks.
2. (a) 5 marks; (b) 15 marks.
3. 20 marks.
4. (a) 6 marks; (b) 5 marks; (c) 9 marks.
5. (a) 5 marks; (b) 5 marks; (c) 10 marks.
6. 20 marks.
7. (a) 8 marks; (b) 5 marks; (c) 7 marks.
8. 20 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) Define the terms cohesion, coupling and adaptability. Explain why maximizing cohesion and minimizing coupling leads to more maintainable systems.

(b) 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, the pump returns it and no fuel is 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.

Using a function-oriented approach, derive a high-level design for the system described above. Make reasonable assumption and clearly state them.

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.

> 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 4. *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 5. *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.

Question 6. *Client-Server Architectures.*

Your customer wants to develop a system for stock information where dealers can access information about companies and can evaluate various investment scenarios using a simulation system. Each dealer uses this simulation in a different way, according to his or her experience and the type of stocks in question. Suggest a client-server architecture for this system that shows where functionality is located. Justify the client-server system model you have chosen.

Question 7. *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?

Question 8. *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.