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 nine 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) 5 marks; (b) 5 marks; (c) 10 marks
2. (a) 10 marks; (b) 10 marks.
3. 20 marks.
4. (a) 5 marks; (b) 15 marks.
5. (a) 10 marks; (b) 10 marks.
6. (a) 10 marks; (b) 10 marks.
7. (a) 5 marks; (b) 15 marks.
8. 20 marks.
9. 20 marks.

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. How is coupling and software portability related?

(b) 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 a function-oriented approach, derive a design for the HSS described above. Make any reasonable assumption and clearly state them.

Question 3. *Function-oriented Design.*

Using a function-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. *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 5. *Software Testing.*

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

(b) 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 6. *Software Reuse.*

(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 their role in supporting code reuse.

(b) Using an *object-oriented* approach, design reusable interfaces for the following abstract data types:

- A stack.
- A character string.

Question 7. *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 8. *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 9. *Software Validation.*

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