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.*

The software life cycle is often compared to the processes involved in building and owning a house. For each stage of the software life cycle, list and describe the comparable processes in the building of a house. How good is the analogy? Are there any stages in the software life cycle that do not have analogies in building a house, or vice versa?

Question 2. *Object-oriented Design.*

Using an object-oriented approach, derive a design for the system outlined below. Make reasonable assumptions about the system when deriving the design.

> 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, it is returned by the pump with no fuel to be 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.

Question 3. *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 an object-oriented approach, derive a design for the HSS described above. Make reasonable assumption and clearly state them.

Question 4. *Software Safety.*

A software system is to be developed for a microprocessor-based *Insulin Delivery System* (IDS) in a hospital. The system works by using a micro-sensor embedded in the patient to measure blood parameters that are proportional the sugar level. These parameters are then sent to a pump controller. This controller computes the sugar level, judges how much insulin is required and sends signals to a miniaturized pump to deliver the insulin via a permanently attached needle.

A low blood sugar level, even for a short term, is a serious condition that can result in brain damage and ultimately death. A high blood sugar level, for a long term, can result in eye damage, kidney damage and heart problems.

(a) Conduct a software hazard analysis of the IDS described above. What are the hazards that can occur in the system? What is the risk associated with each hazard?

(b) Using fault tree analysis, discover the conditions that might cause each of the hazards you identified above.

Question 5. *Software Reuse.*

Software reuse is an important technique of software cost reduction and productivity improvement. Using an *object-oriented* approach, design reusable interfaces for the following abstract data types:

1. A stack.
2. A character string.

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 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 8. *Software Validation.*

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