National Exams May 2012

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 theyappear in your answer book will be marked.

5. 5. All questions have equal weight.

Question 1. *The Software Development Process.*

Compare the life cycle of software to the life cycle of purchasing and owning a piece of equipment, such as a car or a refrigerator. In particular, contrast the life cycle costs of owning the equipment with that of software. How are they similar and how are they different? Justify your answer.

Question 2. *Software Reuse and Portability.*

(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 the pros and cons of using each to support code reuse.

(b) You have been assigned the task of implementing a calendar and clock which gives time and date information. This has to operate on a range of computers from 8-bit micros to 64-bit special purpose processors. Design and implement an abstract data type for representing the calendar and clock that can be readily ported from machine to machine.

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. *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 6. *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) Derive a set of test cases for the following components:

1. a sorting routine which sorts arrays of integers.

2. a routine which takes a line of text as input and counts the number of non-blank characters in the line.

3. a module designed to read in a date expressed using the format YYYY/MM/DD, where YYYY is the year (exactly 4 digits), MM is the month (1 or 2 digits allowed), and DD is the day (1 or 2 digits allowed). Spaces are to be ignored. Thus, valid entries might be 2011/ 1/ 1, 1990 /12/25, or 1250/03/18.

Question 7. *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 8. *Software Validation.*

Explain why single-threaded (i.e., sequential) software designs are easier to validate than designs that involve multiple threads (i.e., parallel).