TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

TU Delft

Exam IN3205-I
Software Quality and Testing
Softwarekwaliteit en Testen

March 26, 2009

- There are 5 questions worth of 18 points in total.
- Total number of pages: 2.
- Use of books and readers is not allowed
- Write clearly and avoid verbose explanations: Points may be deducted for unclear or sloppy answers
- You can answer in English or in Dutch
- Please list your answers in the right order
- The tentative grading scheme is:

Question:	1	2	3	4	5	Total
Points:	3	4	3	5	3	18
Score:						

- If p is the number of points you score, the exam grade E will most likely be determined by

$$E = 1 + 9 * p/18$$

- If the grade for your labwork in 2009 was "good", this can earn you 0.5 extra points.

GOOD LUCK!

1. In Chapter 3, Pezze and Young describe six engineering principles for software analysis and testing.
   (a) (1 point) Explain their *sensitivity* principle.
   (b) (1 point) Explain how this principle is related to the use of *assertions* in, e.g., Java.
   (c) (1 point) Explain how the sensitivity principle applies to the failing radar altitude meter of, e.g., the Turkish Airlines Boeing 737 crashed in Schiphol.

2. After graduating, you start working at a company responsible for processing credit card payments. The documentation turns out to exist of diagrams as shown in Figure 1. This particular diagram describes a part of the authentication procedure.
   (a) (1 point) Turn this diagram into a decision table.
   (b) (1 point) Define the basic condition adequacy criterion. What is the minimum number of test cases needed to meet this criterion for the given decision table?
   (c) (1 point) Define the compound condition adequacy criterion. What is the minimum number of test cases needed to meet this criterion for the given decision table?
   (d) (1 point) Define the MC/DC adequacy criterion. What is the minimum number of test cases needed to meet this criterion for the given decision table?

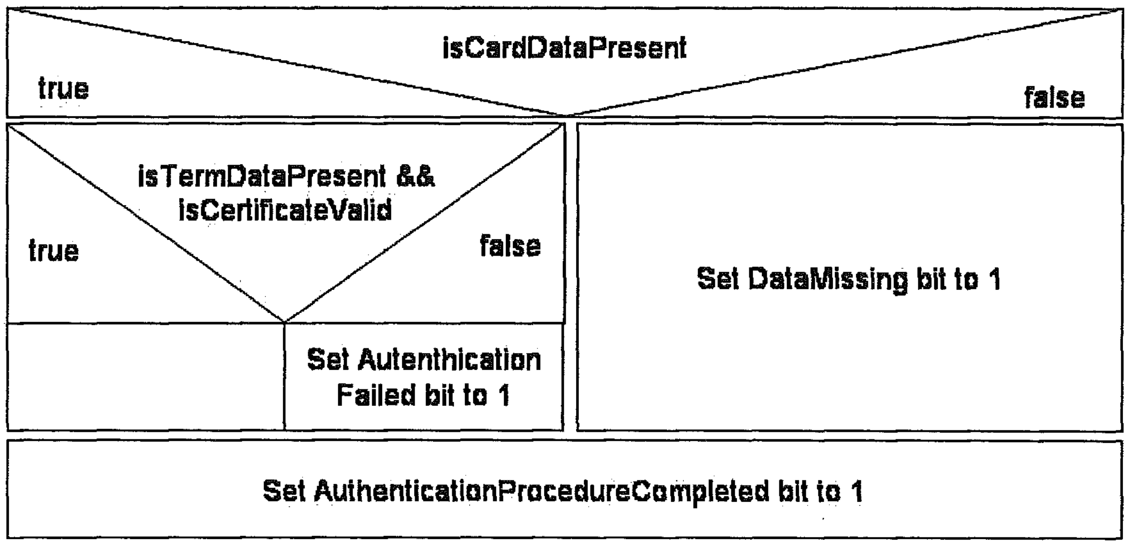

Figure 1: Offline data authentication procedure for credit card payments

3. Automated generation of test data and automated execution of the corresponding test cases is an appealing idea, but requires an *oracle*.
   (a) (1 point) What is an oracle?
   (b) (2 points) Discuss the most important ways of obtaining an oracle, and discuss the advantages and disadvantages of each approach in light of the automated generation of test cases.

4. You are working on the Pacman implementation used in the labwork, and you discover a failure: when restarting the game, the food counter is not reset to zero. Therefore, you start thinking about whether you used the correct adequacy criteria to test the Pacman state machine.
   (a) (1 point) Draw a (UML) statechart diagram for the Pacman game as used during the labwork. Note: the state machine does not have to be exactly as used in the Pacman implementation: all you need is a statemachine that allows you to reason about this fault.
   (b) (3 points) Define the three most important adequacy criteria for testing the given state machine and discuss for each of them why or why it will help you to discover the fault.
   (c) (1 point) Now consider an extension to the Pacman game in which the player has 3 lives, which also results in an adaption of the underlying state machine. Which of your statemachine adequacy criteria do you consider best for testing the correct implementation of this modified state machine?

5. You are involved in testing the implementation of a series of device drivers. For each driver there is a class containing the following method:

   measure(int frequency)

   which conducts a series of measurements with the given frequency. During the measurements a varying number of processors can be used: this is determined by the method *measure*.

   Different groups of device drivers offer different implementations of this method. These implementations differ in: (1) The frequencies accepted; and (2) The number of processors the driver needs to conduct the measurements.
   (a) (1 point) The *measure* method from class D_1 can deal with frequencies 0 to 50 and guarantees that at most 4 processors are needed. Express these properties as pre- and postconditions. Assume there is a method *nrOfProcessorsUsed* which indicates how many processors are actually used.
   (b) (1 point) The tabel below gives the accepted frequences and the maximum number of needed processors for different implementations of the *measure* method in a number of different device drivers.

Class	Frequency	Max. nr. of processors
D_1	0–50	4
D_2	0–50	6
D_3	0–50	2
D_4	0–100	4
D_5	0–25	4

   Which of these classes are LSP compliant subclasses of D_1?
   (c) (1 point) Assume that for each driver D_i a test suite T_i for the *measure* method exists. For which pairs $\langle i, j\rangle$ holds that test T_i should pass for device driver D_j?

End of exam