IN3205
Software Testing and Quality Engineering

IN3205 Example Exam based on earlier IN3420 Exams

March 2008

This exam is *closed book* – no material is allowed to be used during the exam.

You can answer questions in English or in Dutch. For each answer, explain as clearly as possible what you do and why you do it. For unclear or sloppy answers points may be deducted.

There are 15 full questions.

Note that to pass the full exam for this course, you also need to have successfully completed the labwork, the result of which may also be used to adjust your grade.

This trial exam contains representative questions: the actual exam will probably contain fewer questions. It is strongly recommmended to try to make all these questions as a preparation for the exam.

Question 1. You start working for a new company, who are using what they call *transition coverage* as their adequacy criterion for testing statemachines. This coverage criterion requires that each transition explicitly contained in a state machine should be traversed at least once.

An alternative coverage criterion, the all-roundtrip-paths criterion, states that all paths (sequences of transitions) should be followed until either an end state or a state already visited on that path is reached.

Compare these two criteria by answering the following questions:

(1.a) Does one strategy subsume the other? Why?

(1.b) For what sort of faults is transition coverage suitable?

(1.c) For what sort of faults is all-roundtrip-paths coverage appropriate?

You are responsible for testing a new radio/mp3 player to be used in cars. Its specification lists the following features:

- When listening to the radio, pressing the “next” button searches for the next radio station. Pressing the “mp3” button causes the player to start playing the first mp3-song stored in memory.
- When listening to an mp3-track, pressing “next” will skip to the next song. Pressing “radio” returns to playing the last radio station.
- When either listening to the radio station or an mp3-song, a traffic alert may fire. This causes the player to switch to the radio station broadcasting traffic information. As soon as the traffic broadcast is completed the player returns to its previous mode (either mp3 or radio playing).

Question 2. Turn these requirements into a testable UML state diagram.

Question 3. Create and describe a series of test case specications that achieves transition coverage.

Question 4. Identify omitted ⟨state, stimulus⟩ pairs that are ignored, and derive test cases from them (i.e., derive a sneak path test suite).

Question 5. After project completion, you (as project manager) get one of the players for free, and install it in your car. Happily listening to your favorite mp3 songs, a traffic alert message is broadcasted, after which your player continues to play the radio station instead of returning to your song.

You suddenly realize that this is not conform the specifications, and that the player must contain a fault. You start worrying whether your test strategy was properly executed, and, if it was, whether your strategy was capable of finding this fault in the first place.

(5.a) Does the model created in Question 2 contain sufficient information to reason about the causes of this failure? If so, what are possible causes of this failure in terms of that model?

(5.b) Which, if any, of the test cases you listed in the previous questions will necessarily reveal this fault? Why?

Question 6. In addition to that, you discover that when pressing "radio" while listening to an mp3 track, the player appears to return to a random radio station instead of the one tuned in most recently – another deviation from the specifcation.

(6.a) Does the model created in Question 2 contain sufficient information to reason about the causes of this failure? If so, what are possible causes of this failure in terms of that model?

(6.b) Which, if any, of the test cases you listed in the previous questions will necessarily trigger this fault? Why?

You are employed as an engineer for Fortis Insurances, and are working on special loans for students. These are represented by a class *Student* which meet the constraint $18 \leq age \leq 30$, in which *age* is an integer value. The value of *age* can be set in the constructor of the *Student* class.

Question 7. Your system contains different types of students with different invariants. Which of the following classes with given invariants are LSP-compliant subclasses of *Student*, adhering to Meyer's guidelines for subclassing and subcontracting?

Class	**Invariant**
S_1	$0 \leq$ age < 65
S_2	$18 \leq$ age < 27
S_3	$21 \leq$ age < 30
S_4	$16 \leq$ age < 30

Question 8. You are involved in testing the *Student* class and its LSP-compliant subclasses from the previous question. To do this, you decide to create a parallel hierarchy for testing. Provide a UML class diagram of the involved *test classes*.

Question 9. In this setting, you also make use of the *Factory Method* design pattern from the well known book by Gamma and others on design patterns, as done for Pacman in the *createMove* method in the *MoveTest* class that is part of the test suite for the *Move* hierarchy.

Provide the factory method of your class diagram from the previous question, and provide an impementation through the UML note construct.

Question 10. Your colleague Andy starts modifying the Student class, its LSP compliant subclasses S_i and S_j, and the corresponding test classes. You don't want Andy to destroy your carefully designed LSP compliance, so you are watching closely what he is doing.

For each of the three subsequent scenarios, indicate whether you consider it suspicious or safe, and explain your answer.

(10.a) Andy just modifies *Student* and its corresponding test class.

(10.b) Andy only modifies (i) the test class for *Student*, and (ii) the code for S_i.

(10.c) Andy only modifies (i) the test class for *Student*, and (ii) the code for S_i as well as S_j.

Question 11. Provide the smallest control flow graph for which the minimal number of paths required to obtain statement coverage is different from the minimal number of paths required to obtain branch coverage. Provide an example of a corresponding Java program.

You are responsible for developing and testing the software that will be used to rank teams during the UEFA European Championship Football that will be held in Austria/Switzerland in 2008.

In article 6.16 of the official UEFA regulations, you read the following:

> If several teams are equal on points on completion of all the matches in their group, the following criteria will be used to determine the rankings in the order given:
>
> a) Number of points obtained in the matches among the teams in question
>
> b) Goal difference in the matches among the teams in question
>
> c) Number of goals scored in the matches among the teams in question (if more than two teams finish equal on points)
>
> d) Goal difference in all the group matches
>
> e) Number of goals scored in all the group matches
>
> f) Coefficient from the qualifying competitions for the 2004/06 World Cup and 2006/08 UEFA European Football Championship (points obtained divided by the number of matches played)
>
> g) Fair play conduct of the teams (final tournament)
>
> h) Drawing of lots

Question 12. Turn the UEFA requirements into a decision table.

Question 13. Indicate how many test cases there will be in order to achieve MC/DC coverage.

Question 14. You are employed as a senior tester for NuclearPR, the software to be used in the control room of a nuclear power plant. Your manager, mr. Brown, insists that you do exhaustive testing. Write a memo to your manager of at most 150 words summarizing your professional opinion on his vision, offering him advise on what to do.

You are involved in software for electronic payments. Your function should check whether the given credit card data is valid. Data to be take into account include:

- A 16-digit card number. If the card number starts with a 4 it is a Visa card, if it starts with 34 or 37 it is an American Express card.
- An expiry date, which can be at most two years from now.
- A Card Security Code (CSC). For Visa, this code consists of 3 digits, for Amercan Express it consists of 4 digits.

The validation function should conduct basic sanity checks on the card data. These checks include applying a “mod10” algorithm to the card number (the details of the algorithm are irrelevant) which tells if the card number is OK or not, checking that the card is either Visa or American Express, checking that the expiry date has not passed yet, and that a correct number of digits is given for the Card Security Code.

Question 15. Elaborate the category partition testing method for the above setting. To that end, do the following:

(15.a) Identify the function to be tested.

(15.b) Idenitfy the parameters to be tested.

(15.c) For each parameter, determine the categories (characteristics) that should be considered when testing the function

(15.d) For each category, determine the choices (classes of representative values)

(15.e) Identify constraints on combinations of choices

(15.f) Provide actual values for 3 test cases

THE END
