Vrije Universiteit, Department of Computer Science

Examination paper for **Software Testing**
27 March 2014 15:15-18:30

This is a closed book written exam.

No printed material or electronic devices are admitted for use during the exam.

The answers have to be given in English or Dutch.

Both homework and exam are compulsory and graded on an 1 to 10 scale.

The exam grade is calculated as (Q1+Q2+Q3+Q4+Q5 +10)/10.

The final grade is calculated as 0.5*homework + 0.5*exam

A pass is given if both homework and exam components are >= 5.5.

	Q1	**Q2**	**Q3**	**Q4**	**Q5 (code)**	**Σ Qi**	**Maximum credits= (ΣQi+10)/10**
a)	**5**			**5**	**3**		
b)	**5**			**7**	**5**		
c)	**5**				**3**		
d)	**5**				**8**		
e)	**5**				**5**		
f)	**5**						
Total	**30**	**12**	**12**	**12**	**24**	**90**	**10**

Good luck!

Q1. Concepts [30p]

a. Give an example of a test adequacy criterion and explain its role in software testing. [5p]
b. What is an equivalent mutant? Give an example. [5p]
c. Explain the essence of combinatorial testing: why it can be applied and how ? [5p]
d. How does risk-based testing work? [5p]
e. Compare Agile/scrum testing with traditional waterfall testing. [5p]
f. Safety critical software needs a special kind of testing, regulated by standards. Enumerate a few special requirements imposed on testing by these standards. [5p]

Testing from requirements Q2-Q4 [36p]

Q2. [12p]

Consider these requirements for a student examination grading module:

If the student scores 0 to less than 49 then assign D Grade, if the student scores between 50 to 69 then assign C Grade, if the student scores between 70 to 84 then assign B Grade, and if the student scores 85 to 100 then assign A Grade.

Generate test cases using equivalence partitioning combined with boundary value analysis for a by-contract testing of this module.

Q3. [12p]

For a banking software, we want to test a small method, called `validate_withdraw`. This method has to decide whether a withdraw amount, required by the user, will be approved or rejected. A withdraw amount will be approved only if the user has an account that is currently open, it has enough money in it and the current balance left after withdrawal is still more than 100 euro. Both balance and amount have to be positive and cannot exceed 1000000 euros. If any of these conditions is not satisfied, the withdrawal will be rejected

Generate test cases for 1x1 domain testing of these requirements.

Q4. [12p]

These are the requirements for an application that implements an alarm clock:

The user can set the time for alarm to go off.
The user can turn the alarm on or off.
The user can snooze the alarm.
When the clock arrives at the time set for the alarm to ring, then the alarm will ring.

a) Draw a state transition diagram to model this behavior [5p]
b) Apply model based test generation from this state transition diagram [7p].

Q5. Code based testing [24p]

Below is a code snippet that computes the greatest common divisor of two natural numbers by Euclid's algorithm

```
public class Euclid {
    static int gcd(int m, int n) {
        while (m != n) {
            if (m > n) {
                m -= n;
            } else {
                n -= m;
            }
        }

        return m;
    }
```

For this code snippet:
a) Draw the control flow graph. [3p]
b) Generate a test suite that achieves 100% statement coverage. [5p]
c) Enhance your test cases from b) to achieve 100% decision coverage. [3p]
d) Generate a test suite that is adequate with respect to the all-uses criterion [8p]
e) Generate a mutant and show a test case that will kill it. [5p]