Vrije Universiteit, Department of Computer Science

Examination paper for **Software Testing**
28 March 2018 15:15-18:00

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.

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

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

The final grade is calculated as 0.6*homework + 0.4*exam

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

	Q1 (concepts)	**Q2**	**Q3**	**Q4 (code)**	**ΣQi**	**Maximum credits= (ΣQi+10)/10**
a)	**5**	**15**	**15**	**5**		
b)	**5**			**5**		
c)	**10**			**5**		
d)	**10**			**5**		
e)				**5**		
f)				**5**		
Total	**30**	**15**	**15**	**30**	**90**	**10**

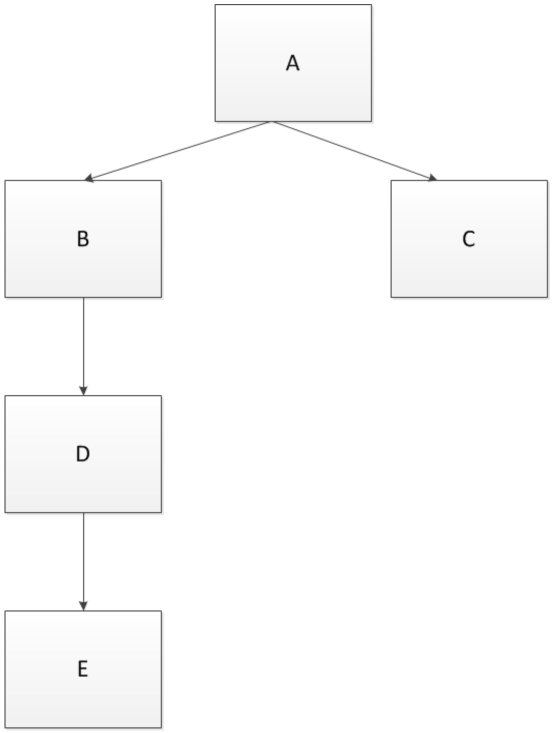
Good luck!

Q1. Concepts [30p]

a. Define the terms *verification* and *validation* and show their place in a software development cycle V-model diagram. [5p]

b. Define the term *metamorphic testing* and give a small example of how it works. [5p]

c. Below is shown the functional dependency graph for 5 software modules. Describe 5 approaches to perform integration testing in this case. [10p]



d. Consider a simplified forward collision avoidance system for cars. The system measures the distance to objects in front using a radar sensor and when the distance is small, it brakes the engine. For this system, identify 5 hazards using one of the hazard analysis methods you know, and for each hazard, generate a mitigation measure and one test scenario. Show in detail how you performed the hazard analysis. [10p]

Testing from requirements Q2-Q3 [30p]

Q2. [15p]

Suppose that you have to test a virtual family doctor software module, that uses the following recommendations for the *Test-med* medicine administration, according to the age of the patient:

- For adults and children older than 12 years: 500 mg
- For children aged 8 to12 years: 375 mg
- For children aged 4 to 7 years: 250 mg
- For children aged 2 to 3 years: 120 mg

Design and generate test cases to defensively test this requirement, by using equivalence partitioning (EP) and boundary value analysis (BVA). Justify your test cases specifications and minimize, if possible, your test cases.

Q3. [15p]

Consider an online store applications and in particular its checkout functionality of processing a shopping cart. The process goes like this. A user fills in the address, then the shipment method, and then she performs the payment. When all the previous steps succeed, the order is complete. Of course the user can decide to change the address, or payment method anytime on the way.

a) Model this system with a state transition diagram.
b) Generate test cases from this diagram. Argument your approach.

Q4. Code based testing [30p]

For the following code snippet:

```
int evensum(int i){
  int sum = 0;
  while (i <= 10) {
    if (i/2 == 0)
    sum = sum + i;
    i++;
  }
return sum;
}
```

a) Draw the control flow graph. [5p]
b) Generate a test suite that achieves 100% statement coverage. [5p]
c) Enhance if needed the test suite from b) to achieve 100% decision coverage. [5p]
d) Generate a test suite that is adequate with respect to the all–uses criterion. Justify your answer. [5p]
e) Generate a mutant and show a test case that will kill it. [5p]
f) Generate an equivalent mutant. [5p]