Vrije Universiteit, Department of Computer Science

Examination paper for **Software Testing**
31 March 2017 12:00-14:45 MF-FG2

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+Q5)*0.9/0.8 +10)/10.

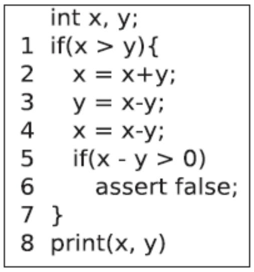
The final grade is calculated as 0.6*homework + 0.4*exam

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

	Q1 (concepts)	**Q2**	**Q3**	**Q4**	**Q5 (code)**	**Σ Qi**	**Maximum credits**
a)	**3**	**9**	**9**	**9**	**3**		
b)	**3**				**5**		
c)	**3**				**3**		
d)	**3**				**3**		
e)	**7**				**7**		
f)	**7**				**3**		
g)					**3**		
Total	**26**	**9**	**9**	**9**	**27**	**80**	**10**

Q1. Concepts [26p]

a. Define the terms ON and OFF point. Give an example. [3p]
b. What is the difference between a verification of a SRS and its validation? [3p]
c. Define the terms safety and reliability and show that they don’t mean always the same. [3p]
d. Define the term Product Risk Matrix and explain its role in testing. [3p]
e. Consider the heater control system built for a neonatal incubator, a rigid box-like enclosure in which a premature born baby can be kept in a controlled environment, for observation and care. The goal is to keep the baby’s temperature at 37°C ± 0.1 °C. Apply the first step of STAMP analysis to this heating control system. Identify one possible accident. Draw a simple control structure for this system. Identify based on this control structure some unsafe control actions. [7p]
f. Explain what symbolic execution is and show how it can be used to generate test inputs for this program: [7p]



```
int x, y;
if(x > y){
  x = x+y;
  y = x-y;
  x = x-y;
  if(x - y > 0)
    assert false;
}
print(x, y)
```

Testing from requirements Q2-Q4 [27p]

Q2. [9p]

Consider the following requirement:

```
[FR 1] The system shall allow shipments for which the price
is less than or equal to €200.
```

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 your test cases.

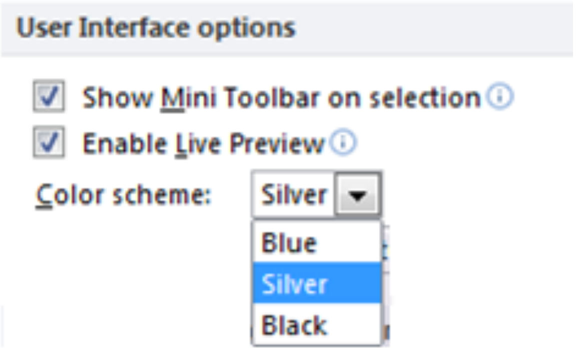
Q3. [9p]

A control system has to count the amount of money dropped into a vending machine. Only 5 and 10 cent coins are accepted. The correct, recognized sum is 25 cents.

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

Q4. [9p]

Given the preferences menu shown below.



User Interface options

☑ Show Mini Toolbar on selection

☑ Enable Live Preview

Color scheme: Silver

a) Identify the inputs. How many test cases do you need to exhaustively test this menu?
b) Design test cases using a pairwise combinatorial model. You should use a suitable orthogonal array from the list in the appendix.

Q5. Code based testing [27p]

For this pseudocode snippet:

```
Read Weight (w)
Read Height (h)
IF w > 400 THEN
    Print “invalid weight”
ENDIF
If h > 3 THEN
    Print “invalid height”
ENDIF
BMI=w/(h*h)
Print ( “BMI = “ BMI)
IF BMI < 20 THEN
    Print “underweight”
ELSE
    IF BMI >= 20 AND BMI <=25 THEN
        Print “ideal weight”
    ELSE
        IF BMI > 25 THEN
            Print “overweight”
ENDIF
```

a) Draw the control flow graph. [3p]
b) Generate a test suite that achieves 100% statement coverage. [5p]
c) Enhance if necessary your test cases from b) to achieve 100% decision coverage. [3p]
d) Draw a data flow graph. [3p]
e) Generate a test suite that is adequate with respect to the all–uses criterion. [7p]
f) Generate a mutant and show a test case that will kill it. [3p]
g) Generate an equivalent mutant. [3p]