Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-01-07
Sal (1)	TER1(26)
Tid	8-13
Utb. kod	TDDD04
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Programvarutestning Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	8
Jour/Kursansvarig Ange vem som besöker salen	Lena Buffoni
Telefon under skrivtiden	013-28 40 46
Besöker salen ca klockan	ja
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, Email: anna.grabska.eklund@liu.se Phone: +46 13 282362
Tillåtna hjälpmedel	Ordbok, Dictionary (printed, NOT electronic)
Övrigt	
Antal exemplar i påsen	

LiTH, Linköpings tekniska högskola
IDA, Institutionen för datavetenskap
Lena Buffoni

Written exam

TDDD04 Software Testing

2019-01-07

Permissible aids
Dictionary (printed, NOT electronic)

Teacher on duty

Instructions and grading
You may answer in Swedish or English.

Your grade will depend on the total points you score on the exam. This is the grading scale:

Grade	3	4	5
Points required	50%	67%	83%

Important information: how your answers are assessed

Many questions indicate how your answers will be assessed. This is to provide some guidance on how to answer each question. Regardless of this it is important that you answer each question completely and correctly.

Several questions ask you to define test cases. In some cases you are asked to provide a minimal set of test cases. This means that you can't remove a single test case from the ones you list and still meet the requirements of the question. Points will be deducted if your set of test cases is not minimal. (Note that "minimal" is not the same as "smallest number"; even when it would be possible to satisfy requirements with a single test case, a set of two or three could still be minimal.)

You may find it necessary to make assumptions in order to solve some problems. In fact, your ability to recognize and adequately handle situations where assumptions are necessary (e.g. requirements are incomplete or unclear) will be assessed as part of the exam. If you make assumptions, ensure that you satisfy the following requirements:

- You have documented your assumptions clearly.
- You have explained (briefly) why it was necessary to make the assumption.

Whenever you make an assumption, stay as true to the original problem as possible.

You don't need to be verbose to get full points. A compact answer that hits all the important points is just as good – or better – than one that is long and wordy. Compact answers also happen to be quicker to write (and grade) than long ones.

Please double-check that you answer the entire question. In particular, if you don't give a justification or example when asked for one, a significant number of points will always be deducted.

1. **Test suite quality (8p)**
    a. Is code coverage a good indicator of test-suite effectiveness? Justify your answer by providing an example of code where coverage alone is insufficient to detect an issue. (4p)
    b. Explain, by illustrating each with an example, how the following can affect the choice of criteria you would use to stop testing. (4p)
        i. Application domain
        ii. Application development process

2. **Basis path testing (20p)**

```
1.  public int foo (int a, int b, int c){
2.      int res;
3.      if (a < 0 || b<0 || c<0)
4.          res = -1;
5.      else if(a>b && a>c){
6.          if(b>c)
7.              res = c;
8.          else
9.              res = b;
10.     } else res = b + c;
11.
12.     return res;
13. }
```

    a. Calculate the cyclomatic complexity for the code above and determine a basis set of paths for the code(6p)
    b. Provide a set of test cases based on the basis path test set you have generated (4p)
    c. Does the test set you created provide **modified condition coverage**? Justify your answer and provide the additional test cases if necessary. (4p)
    d. Explain the principles of **mutation testing**. Provide two examples of **different** mutation operators that can be applied to the code above. Would your test suite detect these mutations? (6p)

3. **Decision table testing (10p)**
    You have the following functionality:
    "A fitness watch uses the following algorithm to set the target daily number of steps for a user: if the user is very active, then set the target at 10000 steps, if the user is moderately active, set the target to 8000 steps, if the user is sedentary, set the target to 6000 steps, if the user is below 30 and not very active, then add 2 thousand steps to the target number of steps."
    a. Draw a decision table for the problem above. If you make any assumptions, state them clearly. (6p)
    b. Generate a set of test cases based on this decision table (4p)

4. **Defect classification (10p)**

a. Explain the difference between faults of omission and faults of commission. Give an example of each. (4p)
b. You are asked to classify the following report using the table below (copy it out on paper first). (6p)

"To conform with the General Data Protection Regulation an encryption module has been implemented in the HR-management tool. When Sara connects to her account now to see her payslip, she gets an "Encryption error" message and no salary information is displayed."

Fault/Defect	Attribute	Value
	Asset	
	Artefact	
	Effect	
	Mode	
	Severity	

5. **Test Planning (10p)**
   a. Describe briefly two activities in the test planning process, for each activity state which artefact it produces (4p)
   b. Define charter and session for exploratory testing (2p)
   c. List two advantages and one disadvantage of exploratory testing (3p)
   d. Give two types of non-functional tests that can be performed (1p)

6. **Integration Testing(4p)**
   a. Give an example of application for which bottom-up integration would be preferable. (1p)
   b. What is the maximum number of stubs needed for top-down integration of a graph with n nodes? (1p)
   c. Explain the principles of MM-path testing and give 2 advantages over top-down and bottom-up testing. (2p)

7. **Software Test Automation (8p)**
   a. What is meant by **sensitive** tests and **robust** tests? Illustrate your answer with an example. (3p)
   b. Explain the role of the adaptor in model-based testing (1p)
   c. Give two advantages and two pitfalls of model-based testing (2p)
   d. How does automated test-case generation influence the test-suite size as a progress indicator? Explain your answer. (2p)

8. Easy Points – True or False (8p)

You get 1p for each correct answer, 0p for no answer, and -1p for each incorrect answer. However, you cannot get negative points for this question.

a. In transition based testing, event coverage subsumes transition coverage.
b. A pragmatic model for testing only includes the aspects of the software relevant to the tested features.
c. Beta testing is unnecessary if installation testing is performed at the customer site.
d. The goal of software testing is to aim for completeness
e. A fault in the program may lead to an error during execution
f. A complete analysis cannot give a false positive
g. The goal of smoke testing is to stress-test a system
h. A thread at system level can be represented as a sequence of MM-paths