



# Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

<b>Datum för tentamen</b>	2011-01-14
<b>Sal</b>	TER4
<b>Tid</b>	14.00-18.00
<b>Kurskod</b>	TDDD04
<b>Provkod</b>	TEN 1
<b>Kursnamn/benämning</b>	Programvarutestning/Software Testing
<b>Institution</b>	<i>IDA</i>
<b>Antal uppgifter som ingår i tentamen</b>	15
<b>Antal sidor på tentamen (inkl. försättsbladet)</b>	7
<b>Jour/Kursansvarig</b>	Kristian Sandahl
<b>Telefon under skrivtid</b>	0706-68 19 57
<b>Besöker salen ca kl.</b>	15
<b>Kursadministratör (namn + tfnr + mailadress)</b>	Gunilla Mellheden, 013-28 22 97, E-post: gunilla.mellheden@liu.se
<b>Tillåtna hjälpmedel</b>	Inga, endast skrivmaterial. No aids are allowed, only writing materials.
<b>Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)</b>	Credits Grade 0-36 U 37-45 3 46-55 4 55-62 5 Exam is shown 2011-02-02 at 11.00-12.30 in room Donald Knuth.
<b>Vilken typ av papper ska användas, rutigt eller linjerat</b>	
<b>Antal exemplar i påsen</b>	

## Exam TDDD04 Software testing 2011-01-14

- Write clearly!
- Please use only one side of each paper and don't address more than one question per page.
- Justify your answers!

### (I) Basic definitions

1. Describe/define the following terms in a software testing context: (4)
  - a. Unit testing
  - b. Integration testing
  - c. System testing
  - d. Acceptance testing
  
2. Describe the difference between the following pairs of concepts: (3)
  - a. Valid input test case – invalid input test case.
  - b. Equivalence class testing – Boundary value testing
  - c. Error of omission – error of commission

### (II) Unit and Integration testing

3. Domain analysis testing (4)

Consider the following example used in a system determining whether to buy shares in a company. The program retrieves a Trade mark rating from 0-10 for the companies. The program also retrieves the data and calculates the P/e-number. The buy algorithm is quite simple: We only buy shares in companies that have:

$8 \leq \text{Trade mark rating} \leq 10$

and simultaneously have:

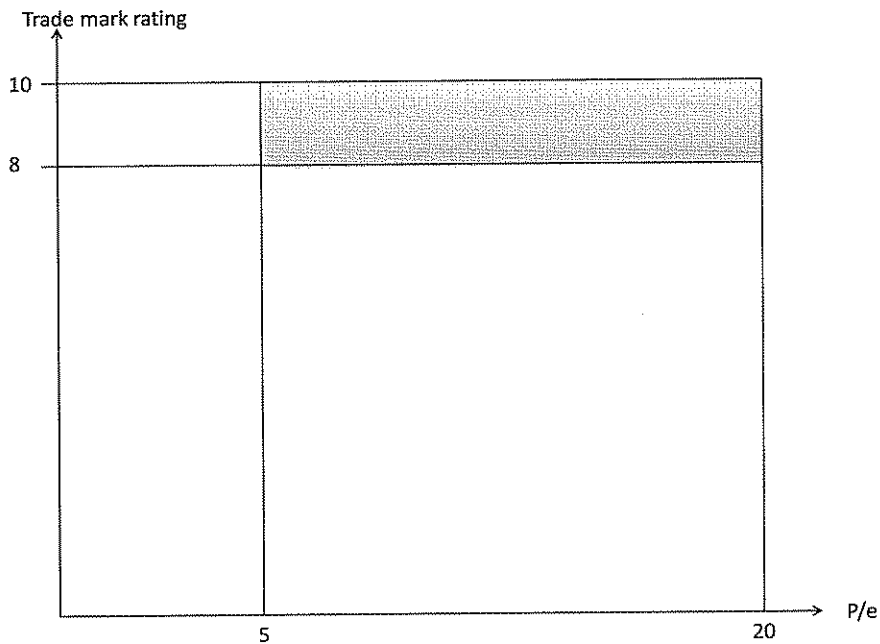
$5 \leq \text{P/e} \leq 20$

This is shown in the shaded area in the diagram below. In all other cases we do not buy.

Your task is now to create a 1x1 domain analysis test case table for the buy algorithm.

Follow the Binder approach which means that you test one **on** point and one **off** point for all conditions of the first variable. Meanwhile the second variable is given typical **in** values. The

you shift so that the first variable holds typical **in** values, whereas the second variable is tested at one **on** point and one **off** point for each condition.



4. Pairwise testing (5)

- a. Which values must A, B, C and D have in the table below to make it a  $L_4(2^3)$  orthogonal array? Don't forget to motivate the answer.

	1	2	3
1	1	1	A
2	1	2	B
3	2	1	C
4	2	2	D

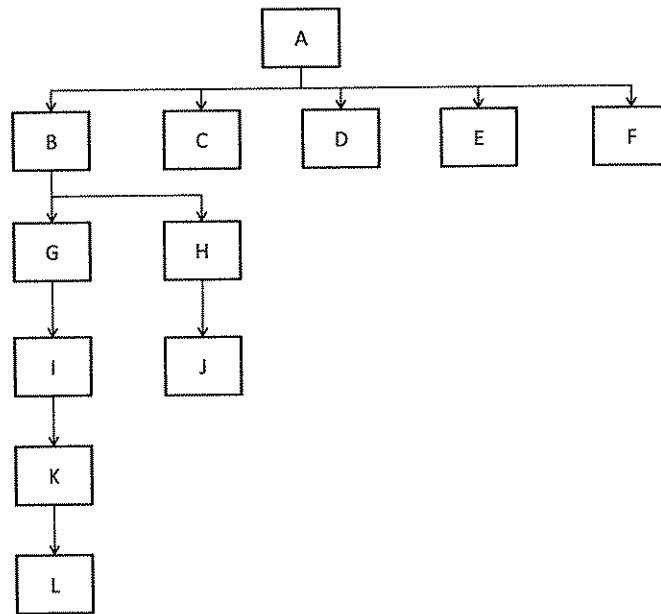
- b. Suppose you have a web-based system which supports the following environments:
- *Browser*: Internet Explorer, Mozilla Firefox
  - *Client operating system*: Windows7, Windows Vista
  - *Server operating system*: Windows, Linux

A system configuration is defined as a set of the three variables, *Browser*, *Client operating system* and *Server operating system*, with valid values. For this system, make a table of inputs of test cases for pairwise testing of the possible configurations with a minimum number of test cases.

- c. Give an argument for using pair-wise testing instead of just generating the same number of random tests.

5. Integration testing. (6)

The following figure illustrates the component hierarchy in a software system.



- Describe the sequence of tests for integration of the components using a bottom-up approach and a top-down approach.
- How many stubs are needed for top-down integration? Don't forget to explain how you calculated the result, since there are different conventions of how to calculate this.
- How many drivers are needed for bottom-up integration? Motivate clearly.

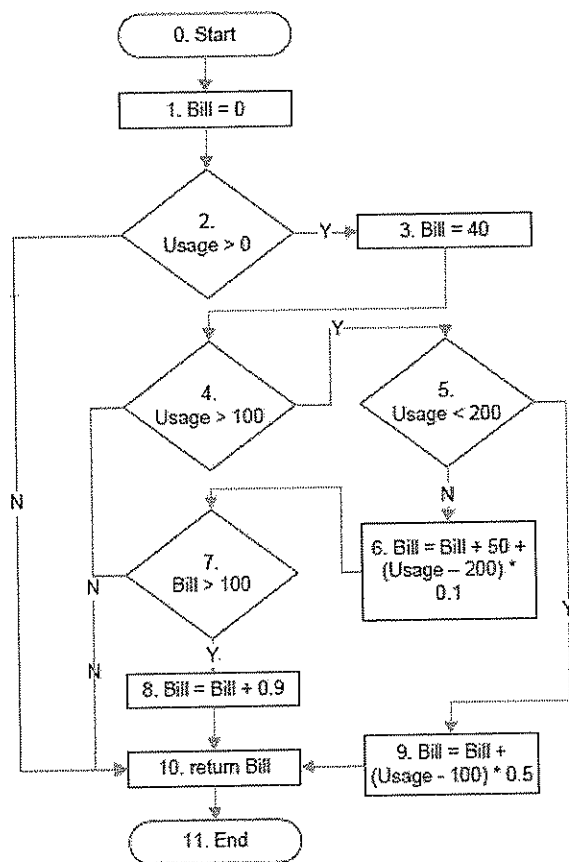
6. Test-driven development (6)

- What are the principle steps of Test-driven development (TDD)?
- Give two potential benefits of TDD
- Give one potential drawback of TDD
- Give an argument for why TDD is so popular in agile methods.

7. What is the difference between stubs and mocks? (2)

8. Data-flow dependencies (6)

Below you find the control flow graph of the billing application in Badlaney et al (2006)

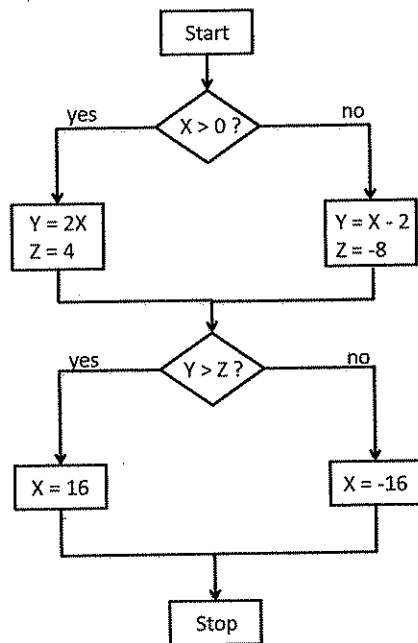


- For the variable "Bill", create a set of test cases that fulfill the All-Uses dynamic data flow testing strategy. For each test case, describe the path taken.
- For the variable "Usage", create a set of test cases that fulfill the All-p-uses dynamic data flow testing strategy. For each test case, describe the path taken.

9. Control flow coverage (5)

Consider the flowgraph below describing a function with an input variable X and output variables X, Y and Z.

- Create test cases for branch coverage using a minimum number of test cases.
- Calculate the cyclomatic complexity the flowgraph.
- Create test cases for basis path testing, using the McCabe method for creating a set of basis paths.



### (III) System testing

#### 10. Cause-effect testing (4)

*Specification:* There are two sensor systems: one for temperature, and one for smoke. If the smoke sensor detects its threshold, the fire alarm starts locally. If the temperature sensor detects its first threshold, T1, the fire alarm starts locally. If both the smoke sensor detects its threshold and the temperature sensor detects T1, then the fire alarm starts at the fire brigade. If the temperature sensor detects its second threshold, T2, the fire alarm starts at the fire brigade. T2 > T1. If only the local alarm is started, it can be stopped if the user enters a PIN-code at the alarm console. If the alarm is started at the fire brigade, it cannot be stopped by the user, neither at the fire brigade nor the local alarm.

Based on this specification:

- Identify causes and effects.
- Design a cause-effect graph for the identified causes and effects. You do not have to model timing, just the logical view of causes and effects.
- Propose a decision table for testing the software.

#### 11. Acceptance test(3)

Describe the following concepts:

- Alpha-test
- Beta-test
- Installation test

12. Path-based integration (3)

Describe the following concepts:

- a. Atomic System Function (ASF)
- b. System thread
- c. ASF graph

**(IV) Automated testing**

13. Model-based testing (5)

- a. Describe a Model-based testing process with at least 3 artifacts and 3 activities
- b. The models must be concise and precise. Describe what these two properties mean, and motivate why this is necessary for good application of model-based testing.

14. Write down a short description of the data-driven scripting techniques. How do you add test cases if you are using a data-driven scripting technique? How do you add test cases if you instead are using a linear scripting technique? (3)

15. According to the IEEE-Std 829-2008 you are recommended to somehow manage Anomaly reports (ARs). In practice this is often done by using a bug-tracking system. Describe three different types of information about anomalies you are recommended by the IEEE-standard to store in such a system. (3)