



TENTAMEN/EXAM

**Kurs/course: Programvaruteknik / Software Engineering
(5DV087)**

Ansvarig lärare/responsible teacher: Jürgen Börstler

Datum/date: 2009-06-12

Tid/time: 9.00-13.00 (4 timmar/hours)

Kod/code: _____

Instructions:

- You can answer in English or in Swedish
- **Write your code and the question number in the upper right of every page**
- Motivate your answers and explain all assumptions
- Please do the following to speed up correction time:
 - Write only on one side of the paper
 - Start a new page for every question
 - Order your papers by question number
 - Mark all questions you have answered
 - Leave a margin for comments

Exercise number	X	Points
1		
2		
3		
4		
5		
6		
7		
8		
TOTAL		

Thanks and good luck, lycka till, viel Glück!

Please mark exercises you have solved in column X.

This page must be returned together with your answers.



Question 1 (4p)

What is an architectural style? Name and describe at least two architectural styles together with their respective advantages and disadvantages.

Question 2 (1+1+2p)

- Explain the design principles of coupling and cohesion.
- Explain why a high degree of coupling in a software design can make maintenance very difficult.
- How could you measure coupling and cohesion? Propose at least one measure for coupling and one measure for cohesion and explain how these measures could be used to determine coupling and cohesion.

Question 3 (1+4p)

- Describe the god class problem.
- Discuss two examples of design heuristics for resolving the god class problem.

Question 4 (3+1+2p)

In the lecture, we have talked about quality factors for requirements (i.e. properties of “good” requirements and “good” requirements documents.

- Name and explain some of them (you get credits for at most six).
- What are non-functional requirements? Please describe.
- Assume you are describing the requirements of a web-based course registration system for Umeå University. Propose some concrete non-functional requirements of reasonable high quality (you get credits for at most four).

Question 5 (4p)

Model a state machine diagram for a microwave oven. Make sure the time can be changed without stopping the oven. Make also sure that the oven always can be opened safely during operation.

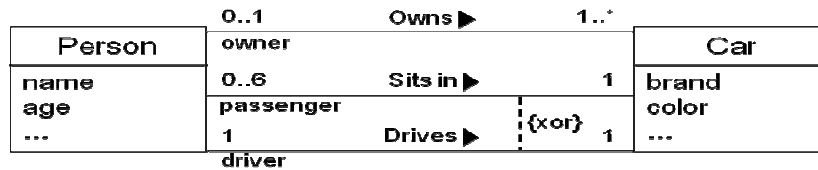
Question 6 (1+1+1p)

- What is the difference between test cases and test data?
- Why is it important to carefully document all tests?
- How does automated TDD support regression testing?



Question 7 (3+2p)

Assume the following UML class diagram describing the relationships between cars and persons:



- (a) Which of the following statements about particular instances can or cannot be correct with respect to the diagram above?
- Jonny and Mary own a red Volvo together.
 - John doesn't own a car.
 - There is always exactly one driver in a car.
 - This silver Saab must belong to somebody.
 - Owners are not allowed to drive a car.
 - Passengers sit in exactly one car.
 - Once Mary drove her Volvo and her Ferrari at the same time.
 - I can take 8 passengers in my Opel.
 - You are not allowed to drive alone.
 - You can either be the driver or the passenger of the same car but not both.
- (b) How would you model the following "problems"?
- The driver must have a driving license.
 - Drivers can only drive their own cars.

Question 8 (1+2+5p)

- What is the main purpose of testing?
- Explain the main idea(s) behind coverage-based testing to develop test cases.
- Develop test cases for reaching branch coverage for the following example:

```

function gcd( x, y: integer): integer;
(* Precondition: x,y > 0 *)
begin
  while x <> y do begin
    if x > y then
      x := x-y
    else
      y := y-x
    end;
  gcd := x
end;
  
```