



Umeå University, SE-901 87 Umeå, Sweden
Department of Computing Science/Jürgen Börstler
Phone: +46 90 786 67 35, Fax: +46 90 786 61 26
E-mail: jubo@cs.umu.se

EXAM 5DV087, VT10

Date 2010-06-11

TENTAMEN/EXAM

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

Ansvarig lärare/responsible teacher: Jürgen Börstler

Datum/date: 2010-06-11

Tid/time: 09.00–13.00 (4 timmar/hours)

Namn/name:

Personnummer:

E-mail:

Kod/code: _____

OBS!

Detta blad kommer att avskiljas innan läraren får skrivningen för rättning.
Ovanstående kod måste därför finnas på samtliga svarsblad när du lämnar skrivningen till skrivvakten. **Memorera ditt kodnummer** eftersom det även kommer att användas som referens när resultaten meddelas.

Please note!

This sheet of paper will be detached before marking. **The code above must therefore be noted on all answer sheets** when you turn in your exam.
Memorize your code, since it will be used when results are announced.

Till skrivningsbevakaren:

Avskilj detta försättsblad och stoppa i kuvert som skickas till **Yvonne Löwstedt**,
Datavetenskap



Umeå University, SE-901 87 Umeå, Sweden
Department of Computing Science/Jürgen Börstler
Phone: +46 90 786 67 35, Fax: +46 90 786 61 26
E-mail: jubo@cs.umu.se

EXAM 5DV087, VT10

Date 2010-06-11



TENTAMEN/EXAM

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

Ansvarig lärare/responsible teacher: Jürgen Börstler

Datum/date: 2010-06-11

Tid/time: 09.00–13.00 (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 (40)		

Thanks and good luck, lycka till, viel Glück!

Please mark exercises you have worked on/solved in column X.

This page must be returned together with your answers.



Question 1 (2+2p)

In the lecture, we have discussed fundamental design principles and architectural styles.

- (a) Discuss some fundamental design principles briefly (you get credits for at most four).
- (b) What is an architectural style? Name and describe an example together with its advantages and disadvantages.

Question 2 (2+2p)

In the lecture, we have talked a lot about software quality and quality assurance.

- (a) Define the terms *software quality* and *quality assurance*.
- (b) Describe how software quality can be measured. What do the measurement results mean?

Question 3 (1+2+4p)

In the lecture, we have talked about design heuristics and design patterns.

- (a) What are the similarities of and differences between design heuristics and design patterns?
- (b) Give examples of a few design heuristics and describe their rationales (goals) briefly (you get credits for at most four).
- (c) Name and describe some design patterns *in detail* (you get credits for at most two).

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).

- (a) Name and describe/explain some of these quality factors (you get credits for at most six).
- (b) Define/describe the term *non-functional requirements*.
- (c) Assume you are developing the requirements document for a web-based system for the booking cinema tickets. Suggest some *concrete* (i.e. specific, not generic) non-functional requirements of reasonable high quality (you get credits for at most four).

Question 5 (2p)

Explain the basic ideas behind earned value management. How does it help in project tracking?

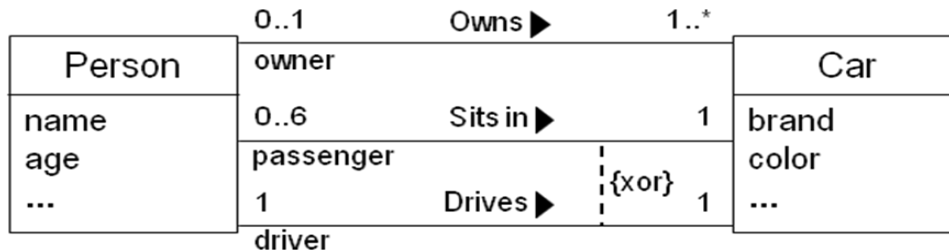
Question 6 (3p)

What are formal inspections? Explain what characterizes formal inspections and how they are carried out.



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+1+5p)

- (a) What is the main purpose of testing?
(b) Explain the main idea(s) behind coverage-based testing to develop test cases.
(c) What type(s) of testing would you argue you are doing in TDD?
(d) 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;
```