



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 2010-04-20

## TENTAMEN/EXAM

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

Ansvarig lärare/responsible teacher: Jürgen Börstler

**Datum/date: 2010-04-20**

**Tid/time: 13.00–17:00 (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



TENTAMEN/EXAM

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

Ansvarig lärare/responsible teacher: Jürgen Börstler

**Datum/date: 2010-04-20**

**Tid/time: 13.00–17:00 (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

*Thanks and good luck, lycka till, viel Glück!*

Exercise number	X	Points
<b>1</b>		
<b>2</b>		
<b>3</b>		
<b>4</b>		
<b>5</b>		
<b>6</b>		
<b>7</b>		
<b>8</b>		
<b>9</b>		
<b>TOTAL (38)</b>		

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 discussed various measures of software quality.

- (a) Discuss the advantages and disadvantages of the measures *absolute number of faults* and *fault density* of a software package.
- (b) Propose and discuss at least two other measures of software quality.

### **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/discuss 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 explain some of them (you get credits for at most six).
- (b) What are non-functional requirements? Please describe.
- (c) Assume you are describing the requirements for a web-based system for booking cinema tickets. Propose some concrete non-functional requirements of reasonable high quality (you get credits for at most four).

### **Question 5 (2p)**

What is the critical path in a project schedule? Why is it important to carefully track progress along the critical path, in particular?



### **Question 6 (1+4p)**

- (a) What is the main purpose of testing?
- (b) We have discussed two main approaches to developing test cases; equivalence-based testing and coverage-based testing. Describe and compare these approaches *in some detail*.

### **Question 7 (3p)**

In the lecture, we have defined TDD (Test Driven Development) by a rhythm, testing patterns, and values. What is meant by these terms? Please describe each of them in some detail.

### **Question 8 (2p)**

Please explain the basic idea behind earned value management. How does it help in project tracking?

### **Question 9 (5p)**

Model a state machine diagram for the Greed game. Please note that not all details from the game description below are relevant for the state machine diagram.

Greed is a dice game played between two or more players. The objective of the game is to register points from the dice rolls. Points are accumulated separately for each player. The game is played with 6 dice. The first player reaching a certain amount of points (usually 10000) wins.

The game is played in turns. In each turn a player can register points which are accumulated through all turns. To register points for a turn a player must reach a certain limit (usually 300 points) on the first roll of every turn, otherwise s/he is considered bust.

If a player goes bust, s/he gets no points for the current turn and must wait for her/his next turn to roll again. The points registered in earlier turns are not affected.

If the first roll of a turn reaches or exceeds the bust limit, the player has two options:

1. to stop and register the points, or
2. to continue and try to get more points in the current turn.

To continue, the player rolls all dice that have not yet scored in the current turn/roll. If no dice are remaining, the player continues by rolling all dice again. The player can choose which dice will score in a roll. This choice must be valid according to the scoring rules. There must be at least one scoring die in each roll.

A player may continue as long as at least one die scores in each roll. If a roll produces no points, the player goes bust and all points made in the current turn are lost. The points scored in earlier turns are not affected.

If a player stops rolling without being bust all points scored in the current turn are added to the scores from earlier turns. If the winning limit is reached, the player has won the game.

Each dice roll is scored as follows:

- Three of a kind score  $100 \times \text{face value}$  of one of the dice. If the face value is 1, then it is scored as 1000.
- Single 1's and 5's score 100 and 50, respectively.
- A street, i.e. 1, 2, 3, 4, 5, and 6 scores 1000.