# Tentamen Software Engineering (400071)
# 22 August 2003

This exam is centered around the case study "LEGO® Animal Builder" (LAB). All questions refer to the LAB case study.
The LEGO company (the toy company "of the building bricks") wants to offer an interactive game called "Animal Builder" on the Internet. Its potential users are children that use LEGO toys and are learning to do interactive games. Therefore, the interface must be colorful, intuitive, simple and attractive.
The LAB allows building an animal out of a set of available building bricks. The user needs first to select the animal she/he wants to build, and then start the game. The graphical interface shows:

- The grid on which building bricks can be placed: the grid is a 3D area in which building bricks can be moved in three directions: up and down; left and right; forward and backward. Note that all directions are defined relative to the animal (e.g. left to the tiger).
- The shape of the animal projected on the grid: the shape shows the types of building bricks the animal is made of. In this way, the user can see which brick to put first.
- The toolbar with all the building bricks available and all the available colors: the user can select a brick and afterwards select the color for that brick. Once selected, the brick appears on top of the grid and the user can start moving it in the three different directions.

During the game session, the user can always select a brick placed in the building area and remove it. In addition, to get some hints, the user can select the "show-me" function, which depicts how the completed animal should look like. The user can print the building area with what has been done, select the help facility, and close the current game session.
The game is freely accessible to everyone on the Internet. However, a possible future extension is to add more features for registered users (i.e. authenticated with username and access code).

An additional service is to switch to the shopping functionality, which allows the user to order online the brick set for the animal she/he played with.

Note: this problem description may be ambiguous and incomplete. In answering the questions, you are free to complete it (if needed) and to briefly motivate your assumptions.

## Question 1: Life cycle models

a) Provide the definition of the following models:
   - Waterfall model
   - Evolutionary prototyping
   - Incremental development

b) For each model, provide a graphical representation, which shows the specific phases and their interactions.

c) Show how these models fit into the spiral model.

d) Choose a software life cycle model for the LAB case study. Motivate your answer.

## Question 2: Requirements engineering

a) Provide the list of functional requirements of the LAB case study, and organize them in a MoSCoW list. Briefly motivate your choices. Also, formalize the "Must have" requirements by means of a Use Case diagram.

b) Provide the list of non-functional requirements of the LAB case study. Motivate briefly each of them.
c) Analyze the LAB case study and provide the class diagram: make use of specialization/generalization (is-a), aggregation (part-of) and/or usage associations. This diagram should provide the conceptual model with emphasis on concepts and existing relationships among them. Class attributes and methods are not required in this diagram. Describe shortly the classes and their associations. If different from UML, explain the notation you used, and motivate your choice.
d) To provide a complete requirements specification for the LAB case study, explain which models should be part of the specification, and (for each model) the diagrams describing them. Motivate your answer.

### Question 3: Software architecture

a) What are architectural styles/patterns? What are design patterns? What is the general structure of a pattern?
b) Design the global software architecture of the system for the LAB case study, and show which patterns you use. Motivate your choices.
c) Explain the relationship between the LAB software architecture and the non-functional requirements stated in the answer for question 2.b).

### Question 4: Software design

a) What is software design?
b) Which is the main difference between requirements engineering and software design?
c) Explain one design criterion.

### Question 5: Testing

a) Explain the difference between error, fault and failure.

### Scoring

With this exam, you can gain 90 points at most. Your final mark is calculated as follows:

Final mark = (#points + 10) / 10

The weight of each question is as follows:

1a: 4 b: 2 c: 4 d: 5
2a: 10 b: 5 c: 15 d: 5
3a: 5 b: 10 c: 5
4a: 3 b: 3 c: 4
5a: 10

### Exam language

If possible, you are kindly requested to write your answers in English. In any case, be sure that your handwriting is clear and understandable.