

# Software Engineering (Adv CS II, 320212)

## Midterm Exam Spring 2006

### Logistics

- You have 120 minutes (sharp) for the test.
- You can reach 76 points if you solve all problems. You will only need 72 points for a perfect score, i.e. 4 points are bonus points.
- Mark all sheets you deliver with your name to make sure it can get graded. We cannot grade if not present or illegible!
- *Take it slow and avoid rushing to mistakes! Different problems test different skills and knowledge, so do not get stuck on one problem.*

**Name:**

*(To be used for correcting, do not write into box below)*

Task	1.1	1.2	1.3	2.1	3.1	4.1	4.2	5.1	5.2	Total
Total	5	3	4	3	9	4	6	1	2	
Reached										

Task	5.3	5.4	6.1	6.2	6.3	6.4	7.1	7.2	7.3	Total
Total	6	5	6	1	3	4	6	2	6	<b>76</b>
Reached										

## 1 General

**Task 1.1 (5 pts):** List 5 elements which are part of a software product, as discussed in lecture.

**Task 1.2 (3 pts):** Name 3 reasons that require software to change over time.

**Task 1.3 (4 pts):** Pressman describes Software Engineering in terms of stacked layers. Name these layers.

## 2 Compile / Link

**Task 2.1 (2+1 pts):** What is the difference between static and dynamic linking? What is the requirement on the target environment, i.e., where the program finally is run, in the case of dynamic linking?

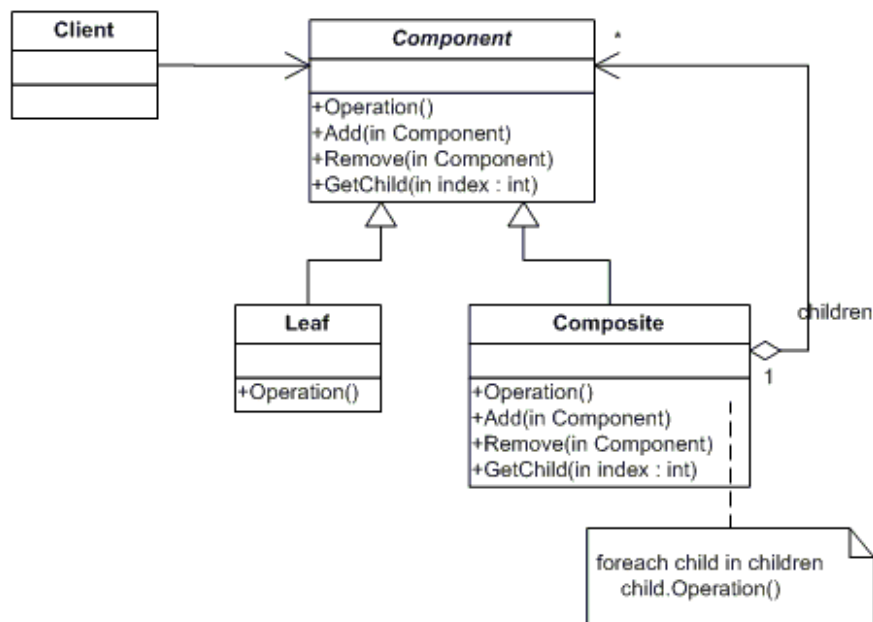
## 3 Defensive Programming

**Task 3.1 (3+3+3 pts):** Name three techniques for error handling in C/C++, explain them briefly, and name one **disadvantage** of each one.

## 4 Design patterns

**Task 4.1 (4 pts):** Give a description of the Singleton pattern in terms of the four elements introduced in lecture.

**Task 4.1 (2+4 pts):** The UML diagram below shows the Composite pattern. Explain its purpose and give an application example by using the pattern's elements.



## 5 Testing

**Task 5.1 (1 pts):** For an average commercial software project, how much effort (in percentage) is appropriate to reserve for testing?

**Task 5.2 (2 pts):** Explain regression testing.

**Task 5.3 (2+2+2 pts):** Give one advantage and disadvantage for each of the following testing methods:

- black box
- glass box
- regression

**Task 5.4 (5 pts):** Coverage testing: for the following code piece, describe a set of (a,b) test cases that altogether will execute every statement at least once.

```
int sign(int a, int b)
{
    int result = 0;
    if (b==0)
        result = 0;
    else if (b>0)
    {
        if (a>=0)
            result = 1;
        else
            result = -1;
    }
    else
    {
        if (a>=0)
            result = -1;
        else
            result = 1;
    }
    return result;
}
```

## 6 Debugging and Bug Tracking

**Task 6.1 (3+3 pts):** Name three different types of illegal memory access (i.e., pointer access) in C/C++ which can lead to program crashes. For each of these, write a short code fragment that produces such a situation.

**Note:** the code fragments do not have to be complete, do not have to compile; syntax is not graded. It is sufficient to display the idea.

**Task 6.2 (1 pts):** What is a memory leak?

**Task 6.3 (1+2 pts):** The following C code has a memory leak; describe it, and change the code in a way that the leak is removed, but the code still produces the same result as before.

```
int countPrimes(int n)
{
    PrimeSet *p = new PrimeSet(n);
    return( p->countPrimeNumbers() );
}
```

**Task 6.4 (4 pts):** On your PC you downloaded and installed a binary program which, however, crashes your machine in some situations. Write a fictitious bug report which is as comprehensive as possible (you may invent situations and details for this purpose).

## 7 Software Configuration Management (SCM)

**Task 7.1 (3+3 pts):** Name three problems related to SCM that can occur when creating and maintaining complex software systems, and explain, for each one, a mechanism of SCM that helps to overcome it.

**Task 7.2 (1+1 pts):** Explain the terms "release" and "configuration".

**Task 7.3 (2+2+2 pts):** Consider the following Makefile contents.

```
all: prog
x.o: x.cpp globals.hh
    $(CC) -c x.cpp
y.o: y.cpp globals.hh
    $(CC) -c y.cpp
prog: x.o y.o
    $(LD) -o prog x.o y.o -lc
```

a) draw the dependency graph

b) Assume that, following a „make all“, `y.cpp` has been changed; what sequence of commands will be generated by repeating "make all"?

c) Answer the (b) question if instead `globals.hh` has been updated.

*– end of exam sheet –*