Submit **THIS FORM** with your answers and on each page your name. Student Name: ……..……

EXAMINATION

Course: Software Management (SMT)
Code: 234004
Example **Date**: June 01, 2006
Exam **Time**: 13:30 – 15:45 **(2 hours + 15 minutes)**

University of Twente
department of computer science

Student Number: **Email:**

Instructions (read them carefully!!)

ONLY the use of a hardcopy of the text book Fenton & Pfleeger (FP) and a hardcopy of the lecture slides are allowed during the examination!
The use of any other resources leads to invalidation of the examination.

Answer the questions on **all** pages! Read the questions carefully. Give answers in the context of this course and the text book.

- For multiple choice questions, select and encircle **all** - one or more - correct answers (see Example Question below). The score (between 0 and 1) is rated according to the formula max(X-Y,0)/N, where X is the number of the selected correct answers, Y is the number of the selected wrong answers and N is the actual number of the correct answers in the question.
- For each question, the maximum number of points (Pts) for a complete correct answer is indicated in the second column.
- The total number of points is 40. You pass the exam with grade 6 when you have a percentage p = 60.0% of the total number of points. The other grades are obtained through interpolation and rounding.
- Check the available time for this exam (see the heading of this page).
- Submit **all** pages of this form with your name on top of **each** page.
- Include clear references to answers on submitted additional papers (also with your name on it)

Example Question *(with encircled correct answers)*

A statement is meaningful (in FP measurement theory) if:

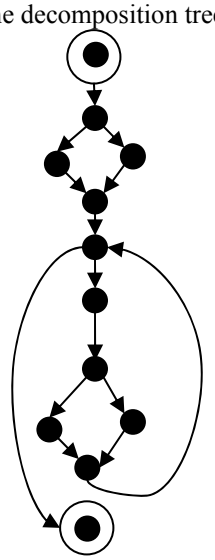
a. the statement is true under admissible scale transformations
b. the statement is false under admissible scale transformations
c. the statement remains true under admissible scale transformations
d. the statement remains false under admissible scale transformations
e. none of above statements is correct

a	b	c	d	e

Submit **THIS FORM** with your answers and on each page your name. Student Name: ……..……

Nr	Pts	**Questions**					
Q1	1	Base Functional Components (in cost estimation) are: a. elementary elements for reuse b. elementary elements in flowgraph decompositions c. elementary elements in functional user requirements d. elementary elements in functional requirements e. elementary elements in non-functional requirements	a	b	c	d	e

Q2	3	Draw the control flowgraph (not compressed - according to the FP rules) of the following program fragment:
		`int n = 10;`
		`System.out.println('n');`
		`for (int i = 1; i < n; i = i + 1) {`
		`  if (i < 5) { System.out.println('n'); }`
		`}`
		`System.out.println('i');`					

Q3	1	The number of edges in the flowgraph of question 2 is: a. 7 b. 8 c. 9 d. 10 e. 11	a	b	c	d	e
Q4	1	The quality of the software development process is the main focus of the standard: a. ISO 9000 b. ISO 9126 c. ISO 12207 d. ISO 15504 e. ISO 15939	a	b	c	d	e
Q5	3	Draw the decomposition tree (with FP rules) of the following control flowgraph: 					

Submit **THIS FORM** with your answers and on each page your name. Student Name: ……..……

<table>
<tr><td>Q6</td><td>1</td><td>The depth of nesting of the control flowgraph in question 5 is:
a. 0
b. 1
c. 2
d. 3
e. ∞</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr>
<tr><td>Q7</td><td>1</td><td>An orthogonal defect classification is a classification that:
a. includes a category with the severity of defects
b. includes a category with the cause of defects
c. has pre-defined categories of defects
d. has mutual exclusive categories of defects
e. has unique categories of defects</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr>
<tr><td>Q8</td><td>3</td><td>The Lack of Cohesion (definition FP) of the following object oriented program fragment is:
…..
<pre>
public class C {
    public C() {
        a2 = 0;
        a3 = 0;
    }
    public void m1() {
        }
    public void m2() {
        a1 = 0;
        a5 = 0;
    }
    public void m3() {
        a2 = 0;
        a5 = 0;
    }
    private int a1;
    private int a2;
    private int a3;
    private int a4;
    private int a5;
}
</pre>
a. 0
b. 1
c. 2
d. 3
e. ∞</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr>
<tr><td>Q9</td><td>3</td><td>The maintainability of 100 modules in two systems has been assessed on an ordinal 5-point scale with frequencies as shown in the table:
<table>
<tr><td></td><td>very poor
1</td><td>poor
2</td><td>moderate
3</td><td>good
4</td><td>very good
5</td></tr>
<tr><td>system 1</td><td>16</td><td>20</td><td>20</td><td>4</td><td>40</td></tr>
<tr><td>system 2</td><td>8</td><td>30</td><td>8</td><td>46</td><td>8</td></tr>
</table>
a. System 1 is more maintainable than system 2
b. System 1 is less maintainable than system 2
c. System 1 and system 2 are equally maintainable
d. The maintainabilities of system 1 and system 2 are incomparable
e. None of these 4 statements (a, b, c, d) is correct</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td></tr>
</table>

LCOM – Lack Of Cohesion Of Methods

LOCOM3 - Lack Of Cohesion Of Methods 3

Measures the dissimilarity of methods in a class by attributes.

Consider a set of m methods, $M_1, M_2, \ldots, M_m$

The methods access a data attributes, $A_1, A_2, \ldots, A_a$

Let $a(M_k)$ = number of attributes accessed by method M_k

Let $m(A_k)$ = number of methods that access data A_k

Then

$$LOCOM3 = \frac{1/a \sum_{i=1}^{a} m(A_i) - m}{1 - m} 100$$

Definition 1. LOCOM3 [Borland Together Architect, version 1.1, 2005]

CBO - Coupling Between Objects

Represents the number of other classes to which a class is coupled. Counts the number of reference types that are used in attribute declarations, formal parameters, return types, throws declarations and local variables, and types from which attribute and method selections are made. Primitive types, types from java.lang package and supertypes are not counted. Excessive coupling between objects is detrimental to modular design and prevents reuse. The more independent a class is, the easier it is to reuse it in another application. In order to improve modularity and promote encapsulation, inter-object class couples should be kept to a minimum. The larger the number of couples, the higher the sensitivity to changes in other parts of the design, and therefore maintenance is more difficult. A measure of coupling is useful to determine how complex the testing of various parts of a design is likely to be. The higher the inter-object class coupling, the more rigorous the testing needs to be.

Definition 2. CBO [Borland Together Architect, version 1.1, 2005]

RFC – Response For Class

The size of the response set for the class includes methods in the class's inheritance hierarchy and methods that can be invoked on other objects. A class that provides a larger response set is considered to be more complex and require more testing efforts than one with a smaller overall design complexity. This measure is calculated as 'Number of Local Methods' + 'Number of Remote Methods'.

Definition 3 RFC [Borland Together Architect, version 1.1, 2005]