



Module Catalog Cohort 2007  
Fontys Hogeschool voor Techniek en Logistiek  
Venlo  
Course SOFTWARE ENGINEERING  
Summary of all module descriptions



# Purpose

This catalog of modules is a compilation of the short module descriptions that were available for the students of cohort 2007.

The overall course plan is given in the table 0.1 on the next page.

At the moment of compilation the students for which this compilation is applicable, have completed all semesters except the last semester (Graduation project in semester 8).

This programme is applicable for the students with student numbers: 2086105, 2116185, 2033046, 2100223, 2116029, 2116435, 2116439, 2117823, 2133563, 2092150, 2109138, 2109142, 2110454, 2114403, 2116438, 2108982, 2109137, 2112154, 2109141, 2111338, 2116434, 2118672, 2109134, 2109136, 2109143 and 2117176.

Venlo, April 12, 2011.

Table 0.1: Curriculum or course plan  
**Major Curriculum Software Engineering – Cohort 2007**

Semester 1: Orientation on I.T.			Semester 2: Business Software Systems		
Code	Name of Module	C	Code	Name of Module	C
PRO1	Programming in Java - 1	6	PRO2	Programming in Java - 2	6
DBS1	Databases	5	SEN1	Software Engineering - 1	3
CSA1	Computer Systems Architecture - 1	6	MOD1	Modelling Techniques - 1	4
MAT1	Mathematics 1	4	MAT2	Mathematics 2	4
PRJ1	Projects 1: <ul style="list-style-type: none"> <li>▪ GUI-Design (3W) 2</li> <li>▪ Office Network (3W) 2</li> <li>▪ Intranet (3W) 2</li> <li>▪ Java/HTML (3W) 2</li> <li>▪ Communication 1 1</li> </ul>		PRJ2	Projects 2: <ul style="list-style-type: none"> <li>▪ Information System - 1 3</li> <li>▪ Information System - 2 6</li> <li>▪ Communication 2 1</li> </ul>	
			ENGL	English for software engineers	3
Sums up to 30 C			Sums up to 30 C		

Semester 3: Technical Software Systems			Semester 4: Advanced I.T. Topics		
Code	Name of Module	C	Code	Name of Module	C
PRO3	Concurrent Programming in Java	5	PRO4	Programming in C/C++	5
MOD2	Modelling Techniques - 2	5	CSA2	Operating Systems	5
PRJ3	Projects 3: <ul style="list-style-type: none"> <li>▪ Lego (7W) 3</li> <li>▪ Lift (7W) 3</li> </ul>		PRJ4	Projects 4: <ul style="list-style-type: none"> <li>▪ .Net Applications (C#)</li> <li>▪ .Net Distributed Applications</li> </ul>	6
MAT3	Algebra	3			
ECO	Economics	3	LINUX	Linux & script languages	4
FND1	Algorithms & data structures	6	FND2	Compilers, scanners & Parsers	5
APPL	Applied research methods	2	SEN2	Software Engineering - 2	5
Sums up to 30 C			Sums up to 30 C		

Semester 5			Semester 6		
Code	Name of Module	C	Code	Name of Module	C
STG1	Practical Period 1 (internship)	30	MIN11	Minor (SOFA)	15
			MIN21	Minor (choice)	15
Sums up to 30 C			Sums up to 30 C		

Semester 7			Semester 8		
Code	Name of Module	C	Code	Name of Module	C
MIN12	Minor (SOFA)	15	STG2	Practical Period 2 (graduation)	30
MIN22	Minor (choice)	15			
Sums up to 30 C			Sums up to 30 C		

# Contents

<b>1 Semester 1</b>	<b>1</b>
1.1 Module CSA1 . . . . .	2
1.2 Module DBS1 . . . . .	3
1.3 Module MAT1 . . . . .	4
1.4 Module PRJ1 . . . . .	5
1.5 Module PRO1 . . . . .	6
<b>2 Semester 2</b>	<b>7</b>
2.1 Module MAT2 . . . . .	8
2.2 Module MOD1 . . . . .	9
2.3 Module PRJ2 . . . . .	10
2.4 Module PRO2 . . . . .	11
2.5 Module SEN1 . . . . .	12
<b>3 Semester 3</b>	<b>13</b>
3.1 Module APPL . . . . .	14
3.2 Module ECO . . . . .	15
3.3 Module FND1 . . . . .	16
3.4 Module MAT3 . . . . .	17
3.5 Module MOD2 . . . . .	18
3.6 Module PRJ31 . . . . .	19
3.7 Module PRJ32 . . . . .	20
3.8 Module PRJ3 . . . . .	21
3.9 Module PRO3 . . . . .	22
<b>4 Semester 4</b>	<b>23</b>
4.1 Module CSA2 . . . . .	24
4.2 Module FND2 . . . . .	25
4.3 Module LINUX . . . . .	26
4.4 Module PRJ4 . . . . .	27
4.5 Module PRO4 . . . . .	28
4.6 Module SEN2 . . . . .	29
<b>5 Semester 5</b>	<b>30</b>
5.1 Module STAGE . . . . .	31
<b>6 Semester 6</b>	<b>32</b>
6.1 Module AMID . . . . .	33
6.2 Module MINOR . . . . .	34
<b>7 Semester 7</b>	<b>35</b>
7.1 Module CMOD1 . . . . .	36
7.2 Module CMOD2 . . . . .	37

7.3	Module GRAP . . . . .	38
7.4	Module JEE . . . . .	39
7.5	Module SOFA . . . . .	40
<b>8</b>	<b>Semester 8</b>	<b>41</b>
8.1	Module AFST . . . . .	42

# 1 Semester 1

## 1.1 Module CSA1

Title	Computer Systems Architectures			
Code	CSA1			
Credits				
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	100			
Description	<p>Theoretical approach of the following subjects:  Transmission media, asynchrone communication, modulation and modems  Packets, packet switched networks, frames, LAN/WAN-technics, routing, protocols and layers  Internet architecture, IP-addresses, IPv6, ARP, ICMP, TCP, UDP  FTP, HTTP, DNS and DHCP</p>			
Literature	Douglas E. Comer; Computer Networks and Internets, 4 <sup>th</sup> Edition; Pearson International Edition, ISBN 0-13-123627-X			
Classroom language	Either Dutch or German			

Note: 1 credit = 28 working hours



## 1.2 Module DBS1

Title	Datenbanken1			
Code	DBS1			
Credits	5			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	15	35		50
Description	The students get to know the architecture and common tasks of database management systems. The following topics are covered in the course: the relational model, entity-relationship modeling, normalforms, database design, relational tuple calculus, SQL, Oracle PL/SQL, triggers, stored procedures, constraints.			
Literature	Ramez A. Elmasri, Shamkant B. Navathe: „Grundlagen von Datenbanksystemen Bachelorausgabe“, Pearson, ISBN 9783868940121			
	Christoph Allen: „ORACLE – PL / SQL für Einsteiger“, Hanser Verlag, Serie „Authorized Oracle Press Editions“, ISBN3446218017			
Classroom language	German			

Note: 1 credit = 28 working hours

### 1.3 Module MAT1

Title	Mathematics 1			
Code	MAT1			
Credits	4			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	40			60
Description	Basics of set theory. (sets of numbers), propositional logic, order of magnitude, real numbers, fractions, summation and product symbols, binomial forms, binomial theorem, factorials, binomial coefficients, square root, powers, logarithms, quadratic equations, inequalities and modulus.			
Literature	<b>Discrete Mathematics (2<sup>nd</sup> edition), Lipschutz (Schaum's Outlines), ISBN 0-07-038045-7</b>			
	<b>Brückenkurs Mathematik (10 Auflage), Bosch (Oldenbourg) ISBN 3-486-25729-3</b>			
Classroom language	German or Dutch			

Note: 1 credit = 28 working hours

## 1.4 Module PRJ1

Title	Project 1			
Code	PRJ1			
Credits	9			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
			70	30
Description	<p>Is divided in 4 subprojects, PRJ11 to PRJ14.</p> <p>PRJ11: Based on a case description the project group (5 students) develop an application based on the GUIDE method. Goal is to implement a prototype of the application graphical interface.</p> <p>PRJ12: Students develop a peer to peer office network based on windows xp platform.</p> <p>PRJ13: Students develop a client server network based on windows server and windows xp clients</p> <p>PRJ14: Students develop a dynamic website based on html, java scripts, and java applets</p> <p>In all these project communication is part of the goals, students have to write reports and have to present there way of working and there results in a power point presentation to the lecturers and the peer groups.</p> <p>Reference code for communication: COM</p>			
Literature	Graphical User Interface Design and Evaluation, D. Redmond- Pyle & A. Moor, Prentice Hall, 1995, ISBN 0-1 3-315193-X			
	„Windows 2000 Server in 24 Hours“, Barrie Sosinsky en Jeremy Moskowit, Sams Publishing, ISBN 0672319403.			
	JavaScript für Einstieger, Martin Bayer, ISBN 87-90785-23-1 Das Einstiegerseminar HTML 4, Thomas Kobert, ISBN 3-8266-7197-X			
Classroom language	German or Dutch			

Note: 1 credit = 28 working hours

## 1.5 Module PRO1

Title	Programming 1: introduction to object-oriented programming			
Code	PRO1			
Credits	6			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	80	20		
Description	<p>This module is an introduction to object-oriented programming (in Java)          To implement the concepts presented the programming language Java and the Java developer's kit BlueJ are used.          The main topics are:          Objects and classes, object interaction, grouping objects in collections, GUI programming (Swing), unit testing, inheritance.</p>			
Literature	Objects First with Java, Barnes & Kölling,			
Classroom language	Native (dutch, german)			

Note: 1 credit = 28 working hours

## 2 Semester 2

## 2.1 Module MAT2

Title	Mathematics 2			
Code	MAT2			
Credits	4			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	40			60
Description	Graph theory, principle of structural induction, geometric series, arithmetic series, matrices, linear equations, solving of linear equations with inverse matrix and Cramer's rule, counting and probability theory			
Literature	Discrete Mathematics Lipschutz (Schaum's Outlines) ISBN 0-07-038045-7 (second edition)			
	Brückenkurs Mathematik Bosch (Oldenbourg) ISBN 3-486-25729-3 (10 Auflage)			
Classroom language	German or Dutch			

Note: 1 credit = 28 working hours

## 2.2 Module MOD1

Title	Object oriented modeling with UML			
Code	MOD1			
Credits	4			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	12.5	75		12.5
Description	<p>Students learn how to use UML and how to design a domain model using UML. Students make a desing and work individually or in a small group, depending on the number of students. This case study starts with a case description and leads to a acceptable class diagram in the first stage of the module. CRC cards are used to obtain the responsibilities for each class. The class diagram has been tested against use case scenario's using sequence diagrams.</p> <p>In the second stage of this module, the design is more implementation oriented. More details are added, state diagrams are used to describe certain classes. Activity diagrams may also be used to get a more complete design. Interaction frames are now introduced in sequence diagrams to show how certain scenario's should be implemented.</p> <p>Important use case scenario's are transformed to concrete test scenario's. Finally a test driven development phase is started. A first implementation in Java is done, based on the test scenario's.</p>			
Literature	Martin Fowler: "UML Distilled", third edition, 2003, Addison Wesley.			
Classroom language	Dutch, English, Germa, as needed.			

Note: 1 credit = 28 working hours

## 2.3 Module PRJ2

Title	Projects 2			
Code	PRJ2			
Credits	10			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
			100	
Description	<p>The students apply their knowledge from the database, programming and modeling courses to develop a small web based information system. They work in groups of 4-6 students. In the first half of the project the students create analysis artifacts (user specification, use cases, domain model, prototypes) and design artifacts (class diagrams, sequence diagrams, ER-diagrams). In the second half of the project, the application is developed using Java web technology (servlets and java server pages (JSP), an oracle database and JDBC to access the database from the web application.</p>			
Literature				
Classroom language	German			

Note: 1 credit = 28 working hours



## 2.4 Module PRO2

Title	Programming 2			
Code	PRO2			
Credits	6			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	15	35		50
Description	<p>Students will gain more insights in Java and how it is applied to practical problems. The following core competences of a software developer are considered: Developing with essential elements of Java, appropriate usage of the programming language, and application of object oriented concepts. Moreover, usage of a professional programming development environment, and the usage of standard libraries is essential. Students will gain basic knowledge of essential features and classes of the Java language. Important concepts will be presented and exercised during practical work in the computer laboratories. The students know how to develop, execute and test within a professional programming environment. The usage of essential libraries is also covered.</p>			
Literature	David Barnes, Michael Kölling: Java lernen mit BlueJ, 2. Auflage, ISBN 3-8273-7152-X			
	Marty Hall, Larry Brown: Core Servlets and Java Server Pages, 2. Auflage, ISBN 3-8272-6954-7			
Classroom language	German			

Note: 1 credit = 28 working hours

## 2.5 Module SEN1

Title	Software Engineering 1			
Code	SEN1			
Credits	3			
Academic year	2007-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	40	60		
Description	<p>A practical approach to testing, in particular unit testing.          We choose this didactic model to both improve student programming skills as well as the software engineering reasons for using (automated) testing.          The module also introduces a few software process aspects and gives practical guidelines to both software creation and version control.</p>			
Literature	Objekt Orientiertes Testen und Testautomatisierung in der Praxis, dPunkt Verlag Heidelberg. ISBN 3-89864-305-0 (for the German class)			
	Object georianteerd Testen en Testautomatisering in de praktijk. Fontys uitgave. (for the Dutch class)			
Classroom language	Dutch and German			

Note: 1 credit = 28 working hours

## **3 Semester 3**

### 3.1 Module APPL

Title	Applied Research			
Code	APPL			
Credits	2			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	12.5	75		12.5
Description	<p>The goal is to learn how to make a structured design for an applied research topic. Research question(s) and subquestions are transformed into more elementary questions about the relevant constructs and aspects. These are the basis for an instrument, e.g. a questionnaire. Students use the theory for a private research case study. With the instrument(s) data will be gathered. The analysis of the results should lead to findings that are grounded. If the design is sound then the original research questions are answered and the results can be easily justified.</p>			
Literature	Nel Verhoeven, Doing Research, isbn: 9789047300649, Boom 2008.			
Classroom language	English			

Note: 1 credit = 28 working hours

### 3.2 Module ECO

Title	Economics for Software Engineers			
Code	ECO			
Credits	3			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	60%	20%		20%
Description	<p>Introduction in economics / business administration.</p> <p>In this module, the following topics will be discussed:</p> <ul style="list-style-type: none"> <li>- Introduction business economics</li> <li>- Legal forms and organizational structures</li> <li>- Porters' Value Chain</li> <li>- Organizational mission and vision</li> <li>- Strategic / tactic / operational planning</li> <li>- Market analysis</li> <li>- Position of the company within its environment / supply chain / Porter 5-forces model</li> <li>- Organization management and management styles</li> <li>- Human resource management / personnel motivation</li> <li>- Marketing</li> <li>- Business- and management information systems / ERP systems</li> <li>- Business and software engineering ethics</li> <li>- The business plan</li> <li>- (Company logistics)</li> </ul> <p>After following the module, the student should be able to describe processes taking place in and around a company and he / she should know what is needed to start his / her own company.</p>			
Literature	Lecture slides			

### 3.3 Module FND1

Title	Algorithms and Data Structures			
Code	FND1			
Credits	6			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	20	30	0	50
Description	Elementary course on algorithms and data structures. Content: algorithms and complexity, big O notation; abstract data types and design issues; datastructures array and linked list, queue and stack, binary tree and tree traversals, priority queue; algorithms divide and conquer, sorting insertion, selection, quicksort, heapsort.			
Literature	Robert Sedgewick: Algorithms in Java, Parts 1-4, third Edition.			
Classroom language	Dutch-German-English			

Note: 1 credit = 28 working hours

### 3.4 Module MAT3

Title	Mathematics 3			
Code	MAT3			
Credits	3			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	40			60
Description	Set theory, relations, functions and algorithms, counting, complexity of algorithms, graph theory, directed graphs, languages, grammars, finite state automata, finite state machines			
Literature	Discrete Mathematics Lipschutz (Schaum's Outlines) ISBN 0-07-038045-7 (second edition)			
Classroom language	German or Dutch			

Note: 1 credit = 28 working hours

### 3.5 Module MOD2

Title	Modeling 2: Reactive systems and patterns			
Code	MOD2			
Credits	5			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	30	40		30
Description	As an extension to MOD 1 we go into the UML way of modeling behavior with state diagrams. The notation for state diagrams and sequence diagrams is completed and several implementation for state machines are introduced (in Java). The second part deals with the Design patterns by Gamma et al. The student investigates several existing projects or API/packages and describes the patterns found.			
Literature	Head First Design Patterns Eric and Elisabeth Freeman, O'Reilly, ISBN 0-596-00712-4			
Classroom language	English			

Note: 1 credit = 28 working hours



### 3.6 Module PRJ31

Title	PRJ31: Lego Mindstorms robot programming			
Code	PRJ31			
Credits	3			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
			100	
Description	<p>This project is an introduction in robot programming. Lego Mindstorms <b>NXTs</b> will be programmed. The programs are developed with Java- Lejos. The main topics are:</p> <p>Sensors, actuators, communication between NXTs (Bluetooth), tasks, implementing robot behavior with the 'behavior control' pattern. In the Robocup tournament teams of students compete against each other.</p>			
Literature	<a href="http://lejos.sourceforge.net/nxt/nxj/tutorial/leJOSNXJTutorial.pdf">http://lejos.sourceforge.net/nxt/nxj/tutorial/leJOSNXJTutorial.pdf</a>			
Classroom language				

Note: 1 credit = 28 working hours

### 3.7 Module PRJ32

Title	PRJ32: Project Elevator in Java			
Code	PRJ32			
Credits	3			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
			100	
Description	<p>In this project apply their knowledge on reactive systems (State machines) and design patterns to a simple elevator system.</p> <p>The students work in groups. Each group is provided with a hardware elevator model which can be controlled with a usb adapter connected to a PC. This connection allows multiple elevators per PC and project. The students also build a GUI simulation model in SWING. This GUI runs in parallel to the hardware and simulates multi cage strategies.</p>			
Literature	Head First Patterns			
Classroom language	English, Dutch, German			

Note: 1 credit = 28 working hours

### 3.8 Module PRJ3

Title	Projects 3			
Code	PRJ3			
Credits	6			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
			80%	20%
Description	<p>This project is sub divided into two parts:  PRJ31- LEGO  PRJ32 – The elevator control  Please refer to the individual descriptions.</p>			
Literature	-			
Classroom language	German, Dutch, English.			

Note: 1 credit = 28 working hours

### 3.9 Module PRO3

Title	PRO3, Concurrency in Java			
Code	PRO3_I			
Credits	5			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	60	40		
Description	<p>Our course in software engineering starts with objects first, then moves towards analysis and modeling using UML and stresses testing during implementation. Our focus is to teach our students programming in the large in stead of programming in the small, by learn them to understand the benefits and consequences of large application programming interfaces. To strengthen this understanding we introduce Design Patterns in semester three as a predecessor module to this module. A natural consequence of teaching software engineering in the above way is that we put less focus on the traditional concurrency problem examples but instead focus of the proper use of parallel programming [Lea] and libraries in the Java world. The Java world provides complete and well understood2 memory models an APIs as building blocks for applications with predictable concurrent behavior. The book Java Concurrency in Practice provides just the right practical approach using modern APIs [Brian Goetz].</p>			
Literature	Brian Goetz, et al: <b>Java Concurrency in Practice</b> . Pearson Education, 2006, ISBN 0-321-34960-1			
	Lea, Doug: Concurrent Programming in Java Second Edition. Addison Wesley, 1999, ISBN 0-201-31009-0			
Classroom language	English			

Note: 1 credit = 28 working hours

## 4 Semester 4

## 4.1 Module CSA2

Title	Operating Systems			
Code	CSA 2			
Credits	5			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	30	40		30
Description	<p>Subject is a general introduction in the working, background and design of operating systems, topics are: overview of computer systems and operating systems, process control, threads, memory management, virtual memory, scheduling strategies, i/o, filesystems.</p> <p>Students practice the theory in practical assignments where they work on a very small educational operating system called Geekos. They implement a loader for a unix elf executable, and implement user threads and system calls.</p>			
Literature	Operating Systemen, vierde editie, William Stallings, Pearson Education Benelux, isbn: 90-430-0695-5			
Classroom language	German or Dutch			

Note: 1 credit = 28 working hours

## 4.2 Module FND2

Title	parsers			
Code	FND2			
Credits	3			
Academic year	2004-2008			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	50	50		
Description	<p>Introduction in parsing.  Subjects: context free grammars, regular expressions, LL(1) parsing, LR parsing, parse trees, visitor pattern, XML parsing</p>			
Literature	R. Hunter - the essence of compilers			
	J. van Loon - on line materials ( Web site)			
	JavaCC - Java parser generator			
Classroom language				

Note: 1 credit = 28 working hours

### 4.3 Module LINUX

Title	LINUX/ Linux and scripting Languages			
Code	LINUX			
Credits	3			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
		100		90
Description	The Linux course is intended to prepare for the LPIC exam LPI101 and has a few extras on Unix/Linux specifics with scripting languages like AWK en (POSIX) regular expressions.			
Literature	<p>* LPI Linux Certification in a Nutshell.  S. Pritchard, B. Gomes Pessanha, N. Langfeldt, J. Stanger and J. Dean:  O'Reilly Media, Sebastopol, CA 954727, 2006. (ISBN 978-0-596-00528-3)</p>			
Classroom language	English			

Note: 1 credit = 28 working hours



## 4.4 Module PRJ4

Title	Projects 4			
Code	PRJ4			
Credits	6			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	10		60	30
Description	In this module the Microsoft .Net technology is introduced. Students will gain knowledge of and get some practical experience with client/server architectures in general, the .Net Framework and its architecture, the various application types: Windows Forms, Web Forms, Windows Services, Web Services, Remoting techniques. Students work together in small groups and make assignments on which they are assessed both theoretically and practically.			
Literature	Reader, Walkthrough, Tutorials, several books recommended (buying not compulsory).			
Classroom language	English			

Note: 1 credit = 28 working hours

## 4.5 Module PRO4

Title	Programming 3			
Code	PRO4			
Credits	5			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	30	40		30
Description	<p>Subject of this course is C/C++ programming and the Qt development frame work. The main topics are: preprocessor macro's, reference and value parameter mechanism, control flow, dynamic memory allocation, structures, arrays, classes, constructors, destructors, inheritance, polymorphism, developing C++ programs based on UML specifications, STL template classes like queues and stacks, Qt dialog based and document view architecture, Qt signal/slot mechanism and event handling, architecture of the Qt frame work, how memory management is done in Qt, thread programming, exception handling, streams, class persistence based on serialization.</p> <p>Students practiced these topics in writing individual there own applications, some examples are: console based text editor (like the old ms-dos ed), dialog based small game, and a cd-information system based on the model/view architecture of Qt</p>			
Literature	Thinking in C++, 2nd ed. Volume 1, Bruce Eckel (=free E-book)			
	Thinking in C++, 2nd ed. Volume 2, Bruce Eckel (=free E-book)			
	Free C/C++ books: <a href="http://www.freetechbooks.com/c-c-f3.html">http://www.freetechbooks.com/c-c-f3.html</a>			
Classroom language	German or Dutch			

Note: 1 credit = 28 working hours

## 4.6 Module SEN2

Title	SEN2 Software Engineering 2			
Code	SEN2			
Credits	5			
Academic year	2008-2009			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	60	40		70
Description	<p>Software Engineering deals with the systematic production of software. It covers methods, techniques and tools for engineering the software, and covers important topics of the development process such as team development, quality assurance and documentation. It contributes to the core topics of the software engineering curriculum. The first quarter practical task is a group wise audit on an existing (but not life) project. The second quarter practical task is the life cycle part of a small project from inception til first delivery.</p>			
Literature	Sommerville, Ian: Software Engineering: (Update) (8th Edition) (International Computer Science Series). Pearson Addison Wesley, 2007, ISBN 0321313798			
Classroom language	English			

Note: 1 credit = 28 working hours

## 5 Semester 5

## 5.1 Module STAGE

Title	Internship			
Code	STAGE			
Credits	30			
Academic year	2009/2010			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
		100		
Description	<p>The student will actually work in a company , foundation or governmental institution for half a year. The student himself has to find a suitable place to do his internship and has to apply for a position. In corporation with the company representative, he or she writes an internship assignment proposal. The company and the assignment proposal have to be approved by one of the lecturers in the university. If the assignment has been approved and the student fulfills all other conditions required to start his or her internship, he or she can start working on his or her internship assignment.</p> <p>A lecturer/counselor from the school will be appointed to the student in order to monitor and guide him in his work and also for judging him or her in the end. The lecturer /counselor will visit the student and the company representative on location on a regular basis.</p> <p>Main objective of this practical semester is to experience the working life in the profession the student is studying for. Another important objective off course is to learn about current working methods, methodologies, practices and technologies.</p>			
Literature	Internship/Graduation manual.			
Classroom language	Not applicable.			

Note: 1 credit = 28 working hours

## 6 Semester 6

## 6.1 Module AMID

Title	Minor Ambient Intelligence and Design			
Code	AMID			
Credits	30			
Academic year	2009-2010			
Education type	Theory (EC)	Practical (EC)	Project (EC)	Self-study (EC)
	12	6	12	
Description	<p>The student acquires knowledge in the field of ambient intelligence, application design and its social implications. The official language used in this course is English.</p> <p>The student is able to:</p> <ul style="list-style-type: none"> <li>• Write a paper about an ambient intelligence subject;</li> <li>• Build a wireless sensor actuator network by means of Arduino Xbee hardware;</li> <li>• Build a small image processing application in MAX-MSP and OpenCV;</li> <li>• Implement a face recognition Java application based on the theory of principle component analysis;</li> <li>• Implement image processing algorithms in Java, e.g. pixel operations, object detection, feature extraction;</li> <li>• Apply different machine learning concepts, such as: statistical modeling, covering rules, association rules, reinforcement learning, instance based learning, clustering, linear models and classification;</li> <li>• Implement an ID3 algorithm, Q-learning algorithm, 2 input - 2 layer – 1 output multilevel perceptron;</li> <li>• Build and program a Lego robot for experimenting with reinforcement learning algorithms, where the robot should be able to learn some predefined behavior;</li> <li>• Work as a team member in a project to realize a prototype of an ambient concept. The deliverables of this project are: poster presentation, live demo of different scenarios, video presentation, software repository, handover document.</li> </ul>			
Literature				
Classroom language	English			

Note: 1 credit = 28 working hours

## 6.2 Module MINOR

Title	Minor (free choice)			
Code	MINOR			
Credits	30			
Academic year	2009-2010			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
Description	<p>Besides some choice modules the bachelor program was fixed until 2005. The major minor model has been introduced in the academic year 2005-2006 to give the student more choice possibilities in his or her study program. Hence, in 2005 two minors of 30 EC and a major of 180 EC were introduced, that together constituted the bachelor program. In 2008 the second minor has been abandoned for practical reasons. Semester 7 was part of the major again, so the major was 210 EC from that time on, the minor of free choice remained 30 EC. Positioned in semester 6.</p> <p>At the time that two minors were in the bachelor program, we positioned the Software Factory, together with communication and two choice modules as a specializing minor. Reference codes: SOFA, COM3B, CMOD2 and CMOD2.</p> <p>In 2008 this combination returned in semester 7 as part of the major. Minors of free choice are chosen by the student from a catalogue with over 40 different minor programs. However, our students mostly have chosen for the minor Ambient Intelligence and design (reference code: AMID) or the International Business Management (reference code: IBMS). In addition students have chosen for a minor abroad, mostly specializing minors at foreign universities (US, Australia, New Sea land, Norway). Due to the great variety in the minors offered, it is not possible to make general statements concerning contents, literature, classroom language and education type. Most minors are described in so called diploma supplements.</p>			
Literature	Not applicable.			
Classroom language	Not applicable.			

Note: 1 credit = 28 working hours



## 7 Semester 7

## 7.1 Module CMOD1

Title	Choice Module 1			
Code	CMOD1			
Credits	5			
Academic year	2010-2011			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
Description	In 2009-2010 the following choice module was programmed and performed:  Reference codes: GRAP			
Literature				
Classroom language				

Note: 1 credit = 28 working hours

## 7.2 Module CMOD2

Title	Choice Module 2			
Code	CMOD2			
Credits	5			
Academic year	2010-2011			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
Description	In 2009-2010 the following choice module was programmed and performed:  Reference codes: JEE			
Literature				
Classroom language				

Note: 1 credit = 28 working hours

### 7.3 Module GRAP

Title	Graphics / Open-GL			
Code	GRAP			
Credits	5			
Academic year	2010-2011			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	20%	60%		20%
Description	<p>This course is an introduction to computer graphics. The focus will be on 3D programming and rendering using the graphics API OpenGL. Topics include creating a display interface, the 3D rendering pipeline, geometric transformations, lighting (pong model), materials, textures, hierarchical modeling, and animation, picking and selection. Extensive programming in C and OpenGL will be required.</p> <p>The final assignment in this course is to develop a 3d-game that has to meet the following requirements:  a landscape existing of material, a landscape existing of textures, differences in height, materials as object facing, spot light which enlightens the objects, ambient light which enlightens the objects, the required objects at least 8, region gaming points, object gaming points, controlled laser visibility, 3 views in one window, Player view, Score/time view, Helicopter view, well-structured source code, well documented, one object with a texture, a moving player, controlled by arrow keys, a shooting player, a menu, mouse events, bezier curves, Picking and selection objects for activation and deactivation, Display lists</p>			
Literature	Textbook: OpenGL a primer, Edward Angel, 2002 Addison Wesley, ISBN: 0-201-74186-5			
Classroom language	English			

Note: 1 credit = 28 working hours

## 7.4 Module JEE

Title	Java Enterprise Edition			
Code	CMOD2			
Credits	5			
Academic year	2010-2011			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
	15	35		50
Description	Students gain basic knowledge of essential features and technologies that are part of the Java Enterprise Edition. Important concepts will be presented and exercised during practical work in the computer laboratories. The students know how to develop, execute and test within a professional development environment. The following topics/technologies will be covered within the module: Enterprise Java Beans 3.0, Architectural Patterns, Development and Deployment of JEE Applications, Basic Application Server configuration			
Literature	B. Burke, R. MonsonHaefel: Enterprise JavaBeans 3.0			
	R. Rahman, D. Lane: EJB3 in action			
	JSR EJB 3.0, <a href="http://jcp.org/en/jsr/detail?id=220">http://jcp.org/en/jsr/detail?id=220</a>			
Classroom language	English			

Note: 1 credit = 28 working hours

## 7.5 Module SOFA

Title	Software Factories			
Code	SOFA			
Credits	18			
Academic year	2010-2011			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
			100%	
Description	<p>“Software factories” is the last project type module in the curriculum. To make it as real life as possible, a real customer is involved. The students work in groups, each group having its own customer and is using product specific technologies. The students take up different project roles which are separately rewarded. The module is concluded with at least one product presentation to the customer and an individual assessment for each student.</p> <p>All documentation is done in English.</p>			
Literature	Sommerville, Software Engineering.			
Classroom language	English, German, Dutch.			

Note: 1 credit = 28 working hours

## 8 Semester 8

## 8.1 Module AFST

Title	Graduation			
Code	AFST			
Credits	30			
Academic year	2010-2011			
Education type	Theory (%)	Practical (%)	Project (%)	Self-study (%)
		100		
Description	<p>The student will actually work in a company , foundation or governmental institution for half a year. The student himself has to find a suitable place to do his graduation assignment and has to apply for a position. In corporation with the company representative, he or she writes an graduation assignment proposal. The company and the assignment proposal have to be approved by one of the lecturers in the university. If the assignment has been approved and the student fulfills all other conditions required to start his or her graduation project, he or she can start.</p> <p>A lecturer/counselor from the school will be appointed to the student in order to monitor and guide him in his work and also for judging him or her in the end. In addition another – independent - representative is appointed to guarantee a certain level of work and an object judging process. The lecturer /counselor will visit the student and the company representative on location on a regular basis.</p> <p>Main objective of this graduation semester is that the student – in a final assessment - proves that he is able to work as a professional in a company or other institution. He or she is encouraged to clearly show all talents and competencies, including reporting and presenting about his or her work. The student has to defend his work in the end for the lecturer / counselor, external representative and company representatives. Another important objective off course is to learn about current working methods, methodologies, practices and technologies.</p>			
Literature	Not applicable.			
Classroom language	Not applicable.			

Note: 1 credit = 28 working hours