

**Faculty of Science**

# **Prospectus 2007 - 2008**

**Computing Science**

Master

**Radboud University Nijmegen**



# Preface

This is the prospectus for the masters programme of Computing Science of the Radboud University. It contains information about the objectives, the goals and the contents of the programme. In addition, it provides a lot of practical information about procedures and organisational matters, as well as a list of important names and dates.

This prospectus has been made with care. Nevertheless, it is possible that it contains some inaccuracies and the authors cannot be held responsible for those. No rights can be derived from the information in this prospectus. Any comments can be sent to Yella Kleijnen, [Y.Kleijnen@cs.ru.nl](mailto:Y.Kleijnen@cs.ru.nl).

July 2007



# List of contents

1	General Information.....	1
	1.1 Introduction.....	1
	1.2 Organisation and practical matters.....	1
	1.3 Administrative details: exams.....	2
	1.4 Internet services.....	3
	1.5 Vademecum and the Statutes for Students at the Radboud University Nijmegen.....	4
	1.6 Important names and addresses.....	4
	1.7 Studying abroad.....	6
	1.8 Calendar 2007-2008.....	7
2	Masters programme.....	8
	2.1 Global structure of the educational programme.....	8
	2.2 Variants.....	8
	2.3 Specialization in Computing Science.....	12
	2.4 Subsidiary Subject.....	15
	2.5 Final thesis.....	15
3	Course descriptions.....	17
	3.1 Course descriptions of the Computing Science courses .....	17
	3.2 Courses MT-variant.....	47
	3.3 Courses C-variant.....	48
4	Practical Information.....	49
	4.1 Important names and addresses.....	49
	4.2 List of lecturers.....	51
	4.3 Procedure for “Schakelverklaringen”.....	52
	4.4 Index of courses.....	53



# 1 General Information

## 1.1 Introduction

The Radboud University Nijmegen is a general university, offering almost all possible academic programmes, ranging from Arts and Law, to Medicine and Science. The Master programmes allow a substantial choice of topics from different areas, thereby offering the possibility of combinations of different studies.

The Institute of Computing and Information Sciences (ICIS) is part of the Faculty of Sciences of the Radboud University and offers a Master of Science programme in Computing Science. This programme constitutes the follow-up to the Bachelor programme of Computing Science, and provides students with a thorough knowledge in computing science.

The Master programme comprises two years. The first year consists of core courses and electives in computing science. The second year is largely devoted to the final thesis work (30 ec), which involves participating in one of the department's advanced research projects or a traineeship or an internship within a company. If you pass the MSc exam you are awarded the degree 'Master of Science'

### Admission

The programme requires a Bachelors degree in Computing Science from the Radboud University, or an equivalent degree. A Bachelors degree in Computing Science from any Dutch university qualifies.

Also students with an post Polytechnic-degree (Dutch: HBO-diploma) can be qualified, although they are always obliged to do an extra, individual programme ("schakelprogramma") of at least 30 ec before entering the master programme as a master student. These students start as bachelor students until they finish their "schakelprogramma". A special intake procedure has been set up for this group of students. Since the "schakelprogramma" consists of courses from the Bachelor programma, it is usually taught in Dutch.

### Credit point system

The Radboud University uses the European Credit Transfer System (ec) employed by all universities in the European Union. One year is 60 ec, the total Master programme of Computing Science comprises 120 ec.

## 1.2 Organisation and practical matters

### Education Bureau

All practical matters regarding the planning of courses and exams as well as various administrative matters regarding students are taken care of by the Education bureau. The staff Members of the education bureau are listed further on in this chapter (see "Important names and addresses"). The coordinator of studies for Computing Science is Yella Kleijnen, Y.Kleijnen@cs.ru.nl. The secretary's office of the Education Bureau can be found in the Huygens building, room HG02.609. General phone number is 024-365 20 84.

## Course programmes and courses

In addition to the information provided in chapter 2, you may find a lot of information about course programmes and courses on the webpages of our institute. The main link for this is <http://www.cs.ru.nl/iii/> where a lot of subpages with practical information can be found.

**N.B. In the course of this academic year, this main link will be changed into <http://www.ru.nl/iii/>; existing pages will be linked automatically to the new URL.**

## Time table and course information

When and where a course or an examination is planned can be found through our time table viewer on <http://www.cs.ru.nl/iii/onderwijs/opleidingen/rooster.html> (English explanation can be found on [http://www.cs.ru.nl/iii/onderwijs/opleidingen/rooster\\_en.html](http://www.cs.ru.nl/iii/onderwijs/opleidingen/rooster_en.html)).

With this time table viewer you are able to compose your personal weekly time table for your individual combination of courses. Indicate the courses that you want to attend and you will be able to view your personal time table for the actual week with a single mouse click.

Via the links on the course names you can find an elaborate descriptions of the individual courses. The list of all courses gives an additional entry: the total of all offered courses, sorted by course name. See <http://www.cs.ru.nl/iii/onderwijs/opleidingen/colleges1.html> (make sure to pick the correct year) .

## Communication

A lot of communication in our institute is done electronically. Lecturers use Blackboard, e-mail or Wiki in their courses. Also, the Education Bureau uses Blackboard and e-mail for important announcements regarding time tables or exams. When you enroll as a student in one of our programmes, you will be added to the Blackboard community for students of our institute. For this, we will use your official RU-e-mail address. *It is your responsibility to make sure that this address always remain intact!*

## 1.3 Administrative details: exams

### Registration for courses and course exams

If you want to take part in a course or an exam, you must register yourself by means of the student internet service system KISS/TIS. More information on this can be found below. *For courses provided by the institute of Computing and Information Sciences (starting with the course code I), a registration for the course is automatically also transferred to a registration for the course exam. Nevertheless you should always check in time whether you have been registered for the exam. If you have not been registered, your grade cannot be administered and you will have to take part in the next examination opportunity.*

### The Master examination

When you have completed all courses of the programme, you qualify for the Master Examination. You have to apply for this examination at the Student Administration /

Examination Office of the Faculty (FSA). To register for this examination, students must submit the following documents:

- valid student card (two cards: registration card **plus** student card. The one is not valid without the other)
- only for students who obtained their bachelor (or propedeuse) certificate elsewhere: bachelor certificate (propedeuse or kandidaats certificate)
- only for students who obtained their bachelor (or propedeuse) certificate elsewhere: an extract from the population register or register of persons, or a copy of the birth certificate
- Only for students who were registered as external students during part of their study: a confirmation of external student status. This is a statement from the institute confirming that the student in question did not receive any education during the period that he/she was registered as an external student.

The Student Administration/Examination Office will only register students for the Master's examination if *all the results* of the interim examinations are in the possession of and have been processed by the Student Administration/Examination Office.

The regulations governing the examinations in August are somewhat different. For these, students can register up to May 30, 2008, and may do so even if several marks have not yet been obtained. These marks have to be delivered before August 29, 2008.

There are 11 examinations scheduled each year (usually the last Friday of the month, except for July). Please check the planning schedule on the notice boards at the FSA. Students should register for the examinations no later than the closing date. The diplomas are presented once every three months. If students need proof of graduation before the date of presentation (e.g. when applying for a job), they can obtain written proof of graduation from the examination committee.

### **Examination Board and Examination Appeals Board**

With regard to examination-related matters, students may first contact the Examination board of the Institute for Computing and Information Sciences (contact details can be found further on in this chapter). In the case of conflict, students can appeal to the Examination Appeals Board of the Radboud University Nijmegen. The procedure to be followed is described in the Vademecum for Students, which can be obtained from the Bureau of Registration of the Radboud University.

## **1.4 Internet services**

### **KISS**

The Radboud University Nijmegen offers all students free access to the Internet and free web mail. Through KISS, students can enroll for courses, sign up for exams, and check their exam results by computer. Every student receives up to 100 MB of free disk space for his or her own website. These 'KISServices' will remain available for at least 6 months after the student has left the Radboud University Nijmegen.

Please note that the KISS password does not give you access to the computers available on campus. For this, the faculty will supply you with a separate pass word.

*Opening hours KISS helpdesk*

Mondays-Fridays: 10 A.M.-5 P.M.

(closed on the first Friday afternoon of each month)

## **Blackboard**

The KISS password you receive will also give you access to the Radboud University digital learning environment system *Blackboard*. Lecturers use Blackboard to supply information about their course, send announcements etc. Blackboard is also used by the Education bureau of the Institute for Computing and Information Sciences to communicate important information to students. Upon registering as a student, you will be enrolled in our community of ICIS-students with your official RU-e-mail address. *Please make sure that this remains intact always.*

## **1.5 Vademecum and the Statutes for Students at the Radboud University Nijmegen**

The Vademecum for students studying at the Radboud University Nijmegen contains general information concerning housing, health care, government funding, studying, students' rights and responsibilities, services, and student associations and organisations. You will also find useful addresses and telephone numbers in this brochure. The information in the Vademecum is also on the Internet: [www.ru.nl/studentenzaken/](http://www.ru.nl/studentenzaken/).

The student statutes consist of a description of the rights and responsibilities of all students registered at the Radboud University Nijmegen, based on statutory and university regulations. The student statutes and its appendices are on the Internet: [www.ru.nl/studentenzaken/](http://www.ru.nl/studentenzaken/). The Vademecum is available free of charge at the Student Affairs Desk, all Lecture Note Centers and the Advice Office for the Faculty of Arts.

## **1.6 Important names and addresses**

### **Faculty of Sciences**

Toernooiveld 1, 6525 ED Nijmegen

Huygens building

tel.: 024-3616161 (Radboud University - general phone nr.)

### **Education Bureau for Computing and Information Sciences**

*General:*

- Secretary's office: HG02.609, Huygens building  
tel.: 024-3652084

*Staff:*

- *Ms. Resi Westerman, MA*, pr/secretary; R.Westerman@cs.ru.nl
- *Ms. Doris Meier, MA*, pr/secretary; D.Meier@cs.ru.nl
- *Ms. Vera Kamphuis, MA*, head, coordinator of studies of Information Science; V.Kamphuis@cs.ru.nl
- *Ms. Yella Kleijnen*, coordinator of studies of Computing Science; Y.Kleijnen@cs.ru.nl

**Student advisor for Master students**

- *dr. Theo Schouten*, T.Schouten@cs.ru.nl

**Student advisor for "HBO-instromers" (post-Polytechnic bachelor students)**

- *Dr. Hanno Wupper*, H.Wupper@cs.ru.nl

**Master's thesis coordinator**

- *Dr. Patrick van Bommel*, P.vanBommel@cs.ru.nl  
website: [www.cs.ru.nl/mt/](http://www.cs.ru.nl/mt/)

**Education Board**

- *Dr. Erik Barendsen*, director; E.Barendsen@cs.ru.nl
- *Dr. Sjaak Smetsers*, coordinator of the master programme Computing Science; S.Smetsers@cs.ru.nl
- *Prof.dr. Erik Proper*, coordinator of the master programme Information Science; E.Proper@cs.ru.nl
- *Freek van den Berg*, studentassessor; F.vandenBerg@student.ru.nl
- *Ms. Vera Kamphuis, MA*, head of the education bureau; V.Kamphuis@cs.ru.nl
- *Ms. Yella Kleijnen*, secretary, Y.Kleijnen@cs.ru.nl

**Education Committee of Computing Science and Information Science**

Members of this committee are 4 students in computing science, 4 students in information science and 4 lecturers.

- *Drs. Ger Paulussen*, chairman; G.Paulussen@cs.ru.nl
- *Ms. Yella Kleijnen*; secretary; Y.Kleijnen@cs.ru.nl

**Examination committee**

- *Dr. Marko van Eekelen*, chairman; M.vanEekelen@cs.ru.nl
- *Dr. Dick van Leijenhorst*, vice-chairman; D.vanLeijenhorst@cs.ru.nl
- *Dr. Theo Schouten*, secretary; T.Schouten@cs.ru.nl

General e-mail account: [examencommissie@niii.ru.nl](mailto:examencommissie@niii.ru.nl)

Website: <http://www.cs.ru.nl/iii/organisatie/gremia/excie/index.html>

### **Coordinator of international affairs for Computing and Information sciences**

- *Dr. Janos Sarbo*, [J.Sarbo@cs.ru.nl](mailto:J.Sarbo@cs.ru.nl)  
Website: <http://osiris.cs.kun.nl/~janos/BUIT.html>

### **Office of administration and exams for students (FSA)**

- Ms. Clementine Hendriks, Ms. Yvonne Mulder,  
opening hours: Monday to Thursday: 13-16 hrs, Friday: 9-12 hrs  
room: HG00.134, Huygens building  
tel.: 024-3652247/024-3653392

### **Student affairs office**

- Comeniuslaan 4, Nijmegen  
tel.: 024-3612345  
webpage: [www.ru.nl/students/](http://www.ru.nl/students/)  
See the 'Vademecum' for more information.

### **Students' association Thalia (for students of Computing Science or Information Science)**

- [info@thalia.nu](mailto:info@thalia.nu) (general info)  
website: [www.thalia.nu/](http://www.thalia.nu/)

### **Alumni association Ninja (for Computing and Information Sciences)**

- Contact: dr. Dick van Leijenhorst, [D.vanLeijenhorst@cs.ru.nl](mailto:D.vanLeijenhorst@cs.ru.nl)  
website: [www.cs.ru.nl/ninja/](http://www.cs.ru.nl/ninja/)

## **1.7 Studying abroad**

In the past years, more and more students decided to visit a foreign university for some time during their study. This may include taking courses, working on a project, or attending a summerschool. There are various possibilities for getting a scholarship, e.g. via the Erasmus program (Europe) and the ISEP program (USA). Within the Erasmus program, our institute has relations with universities in different countries such as Sweden, Denmark, Spain, Portugal and Hungary, but a scholarship in other countries is possible as well.

Via a scholarship you may study abroad for a period of 3 to 12 months. Credits (ECs) obtained abroad can usually be acknowledged by the sending university in the Netherlands. In this matter the Examination board can help you select courses that also satisfy the conditions set by our university. You may also consult the coordinator of external relations at ICIS, J. Sarbo ([janos@cs.ru.nl](mailto:janos@cs.ru.nl)), who can help you set up communication with another university. For matters concerning an application for a scholarship, you may contact the External Relations Office of the Radboud University ([www.ru.nl/er](http://www.ru.nl/er)).

## 1.8 Calendar 2007-2008

### Academic year:

September 1st 2007 - August 31 2008

### Semester:

Fall semester: 3 September 2007 - Friday 1 February 2008;

Spring semester: 4 February - 11 July 2008

### Holiday(s)

During holidays there are no lectures, but it is possible that exams and preliminaries are scheduled during holidays. For more information you may consult the schedule below.

Holiday	Date	Lectures	Preliminaries (tentamens)	exams
Start academic year	afternoon 03-09-07	-	-	-
Fall Break*	22-10-07 / 26-10-07			
Christmas holidays	24-12-07 / 04-01-08	-	-	-
Carnival	04-02-08 / 08-02-08	-	+	-
Easter	21-03-08 / 24-3-08	-	-	-
May holiday	28-04-08 / 05-05-08	-	+	-
Queen's Birthday	30-04-08	-	-	-
Ascension day	01-05-08	-	-	-
day after Ascension day	02-05-08	-	-	-
Liberation Day	05-05-08	-	-	-
Whitsun	12-05-08	-	-	-
Foundation day	15-05-08	-	-	-
Summer holidays	14-07-08 / 29-08-08			
	14-07-08 / 08-08-08	-	-	-
	11-08-08 / 22-08-08	-	+	-
	25-08-08 / 29-08-08	-	+	+

\* Fall break: this break applies for this faculty but is not a general holiday for the Radboud University

### Quarters

Quarter 1: 03-09-07 till 09-11-07

Quarter 2: 12-11-07 till 01-02-08

Quarter 3: 11-02-08 till 18-04-08

Quarter 4: 21-04-08 till 11-07-08

## 2 Masters programme

### 2.1 Global structure of the educational programme

All of the faculties of the Radboud University have implemented the bachelor-master structure. As the same structure has been implemented in most European countries, it is much easier to compare the university training programmes and it is easier to switch between universities. The academic programmes are made up of two components:

- Bachelors programme
- Master's programme

The bachelor takes 3 years, the first of which (propedeuse) concerns foundation courses. The bachelor programme is broadly based and it prepares you for the master programme. After completing this programme you will receive the bachelor's degree, at which time you may call yourself *Bachelor of Science (BSc)*.

The master takes 2 years. You will specialize in one of the research related themes of the Institute of Computing and Information Sciences. Upon completing your studies, you will receive your master's degree and you may call yourself *Master of Science (MSc)*.

The Master programme consists of various components: Basic courses, Variant related subsidiary subjects (variant related), specialization (24 ec), Research and Development (20 ec), free space (variant related), final thesis (30 ec).

A student chooses one of the offered variants described below. The master programme and the size of the components are variant related.

### 2.2 Variants

The Faculty of Sciences offers four variants which prepare students for their future profession. There are four variants from which students choose one:

1. *Communication variant (C)*: trains students in the direction of science communication, internally or towards society.
2. *Research variant (O)*: trains students for fundamental and applied research. This variant is required for people pursuing a PhD position or a position in industrial or institutional research.
3. *Business and Management variant (MT)*: prepares students for a management position as an academic professional. It prepares students for a career in science related business and administration and for innovation and enterprise from an academic perspective.
4. *Educational variant (E)*: prepares students for a career in teaching. *This variant is not yet available for students in computing science.*

Within the chosen variant the student chooses his master programme in Computing Science. Part of the master's programme is a specialization of 24 ec. For the benefit of the student, the specialization courses are grouped in so called themes (see further on). The final project (30 ec) includes a Master thesis and an oral presentation.

**Research Variant (O)**

The Research variant leads students to a high level of knowledge in Computing Science. It consists of advanced courses and a final research project including a masters thesis and an oral presentation of it.

<b>mandatory courses</b>			
<i>Year 1 Fall semester</i>	<i>ec</i>	<i>Year 1 Spring semester</i>	<i>ec</i>
R&D: Research 1	8	R&D: Research 2	6
		Introduction to Philosophy	3
<i>Year 2 Fall semester</i>	<i>ec</i>	<i>Year 2 Spring semester</i>	<i>ec</i>
R&D: Research 3 <i>or</i> R&D: System Development Research	6	Final Thesis	30

<b>Optional subjects year 1-2</b>	<b>ec</b>
specialization/theme	24
subsidiary subject	25
extra courses in Computing Science	24
free choice	6

### Management Variant (MT)

This variant is intended as a preparation for a job in the field of management and a career in science-related business and for innovation and enterprise.

<b>mandatory courses</b>			
<i>Year 1 Fall semester</i>	<i>ec</i>	<i>Year 1 Spring semester</i>	<i>ec</i>
R&D: System Development	8	R&D: System Development Management 1	6
Business and Society		Introduction to Philosophy	3
<i>Year 2 Fall semester</i>	<i>ec</i>	<i>Year 2 Spring semester</i>	<i>ec</i>
R&D: System Development management 2	6	Final Thesis	30

<b>Optional subjects year 1-2</b>	<b>ec</b>
specialization/theme	24
extra courses in Computing science	12
Subsidiary subject Management	25
free choice	6

Subsidiary subject in Management

- Business and Society, 5 ec
- Organization Theory, 5 ec
- Finance and accounting, 5 ec
- Strategy and Marketing, 5 ec
- Innovation management, 5 ec

**Communication Variant (C)**

This variant is intended for a job in communication and not as a preparation for admission as a research PhD student in Computing Science. More information can be obtained at: [www.filosofie.science.ru.nl](http://www.filosofie.science.ru.nl)

<b>mandatory courses</b>			
<b><i>Year 1 Fall semester</i></b>	<b><i>ec</i></b>	<b><i>Year 1 Spring semester</i></b>	<b><i>ec</i></b>
R&D: System Development (version 1) OR R&D: Research 1 (version 2)	8	R&D: System Development Management 1 (version 1)	6
		Introduction to philosophy	3
<b><i>Year 2 Fall semester</i></b>	<b><i>ec</i></b>	<b><i>Year 2 Spring semester</i></b>	<b><i>ec</i></b>
R&D: System Development Research (version 2)	6	Final Thesis	30

<b>Optional subjects year 1-2</b>	<b>ec</b>
specialization/theme	24
Extra courses in Computing science	12
Subsidiary Subject Communication	21
Extra courses in Communication	6
free choice	10

Subsidiary subject Communication (21 ec):

- Introduction Science Communication (3 ec)
- Science & Social Interaction (3 ec)
- Risk Communication (3 ec)
- Framing Knowledge (3 ec)
- Knowledge Society (3 ec)
- Boundary Work (3 ec)
- Science, Media and Strategy (3 ec)

## 2.3 Specialization in Computing Science

Within every variant a student chooses one of the offered specializations of 24 ec. These specializations are directly related to the research groups organized in the Institute for Computing and Information Sciences. These research groups and their fields of interest are:

- **Foundations:** To develop and study formal languages and logical theories involving algorithms, proofs, processes, computations, correctness and complexity, as well as developing and experimenting with computer tools that support these languages and logics.
- **Software technology:** To develop theory, methods and tools for specification, programming (in particular functional and generic programming techniques), static analyses (especially type systems), and dynamic analyses (with a focus on specification and model-based testing) to support designers and developers in the construction and verification of reliable software.
- **Information and knowledge systems:** To perform fundamental and applied research around the theme of knowledge-intensive systems, i.e. systems that can elicit, structure and process implicitly and explicitly represented knowledge of a problem or domain, drawing upon ideas, methods and techniques from information systems and artificial intelligence.
- **Security of systems:** To develop theories, formal methods and tools that contribute to the security of protocols and software, and contribute to ongoing developments and debates on socially relevant issues such as privacy, open source, electronic voting, biometric passports, etc.
- **Informatics for technical applications:** To carry out fundamental research on formal methods and tools for the specification, design, analysis and testing of computer systems for technical applications (in particular embedded systems and protocols), and demonstrate and assess the effectiveness of using these methods and tools in the industrial software development process, notably band structure calculations where Nijmegen is the home of a national facility

The ICIS also offers a *two-year international programme in computing science* at the Kerckhoffs Institute. You will find a description of this programme after the list of specializations.

### Specialization/Theme

Courses are grouped around a theme/specialization. Below you will find an overview of the themes with courses:

#### *Theme Foundations*

- Proof Assistants (6 ec)
- Theory of Complexity (6 ec)
- Semantics (6 ec)
- Type theory (6 ec)

#### *Theme Embedded Systems*

- Testing techniques (6 ec)
- Analysis of Embedded Systems (6 ec)
- Design of Embedded Systems (6 ec)
- Image processing (6 ec)

*Theme Information Systems*

- Business Rules (6 ec)
- Information Retrieval (6 ec)
- Foundation of Information Systems (6 ec)
- Cognition and representation (6 ec)

*Theme Software Construction*

- Reliability of Software Systems (6 ec)
- Advanced Programming (6 ec)
- Compiler Construction (6 ec)
- Testing techniques (6 ec)

*Theme Quality of Software*

- Software Security (6 ec)
- Reliability of Software (6 ec)
- Testing techniques (6 ec)
- Analysis of Embedded Systems (6 ec)

*Theme Artificial Intelligence*

- Computational Intelligence (6 ec)
- Information Retrieval (6 ec)
- Pattern Recognition (6 ec)
- Cognition and Representation (6 ec)

**Computer Security: Kerckhoffs Institute**

Together with the University of Twente and Eindhoven University of Technology, we offer a 2-year international program in computer security.

Our programme will give you a broad and strong background on many aspects of computer security. Towards the end of the programme, you will choose a particular specialization topic for in-depth study. You could, for example, become a smart card expert, a network security and intrusion detection specialist, a cryptographer, or an information security analyst. Developing secure systems and analysing the security of existing systems is a challenging task. Computer security is a topic of growing importance, as it affects ever more aspects of business, government, and our daily life as citizens. That is why security specialists are in high demand. The Kerckhoffs Master in Computer Security offers excellent career prospects in both the public and the private sector, and in academic as well as industrial research.

After completing the master program, you are ready to take on one of the following job opportunities.

- Security architect
- Security consultant
- Security policy maker
- Industrial researcher
- Academic researcher (towards international PhD)

The master programme covers two years, divided into two semesters each. Each semester offers two mandatory courses, and a selection of optional courses.

The courses cover a wide range of topics, including but not limited to: cryptography, system/network security, organizational security, privacy, data management. Courses are taught in English. Course materials are in English as well.

The last semester is reserved for the graduation project and writing a master thesis. Master projects can be either internal, or external at a company for instance.

Courses are taught in Nijmegen, as well as the University of Twente and Eindhoven University of Technology. Lectures at Enschede and Eindhoven are scheduled exclusively on Mondays and/or Fridays. Students need to travel one day a week on average. Optional courses are scheduled back to back with mandatory courses to minimize such travel.

In Nijmegen, the Kerckhoffs Programme is embedded in the regular Computer Science Master. On successful completion of the program, you will obtain a master diploma in Computer Science, with a certificate attesting you specialized in Computer Security at the Kerckhoffs Institute.

The Security of Systems group of the Radboud University Nijmegen works at the forefront of computer security research. We investigate new security vulnerabilities, build tools to analyse the security of critical systems, and develop new, stronger, countermeasures. We are also actively involved in the lively debate on the effects of security measures on society at large, such as the introduction of biometric passports, or electronic voting.

#### *Admission requirements; procedures*

To register at the Kerckhoffs Master programme, you need to first register as a full time computer science master student at the Radboud University Nijmegen. After that, you also need to register yourself at the Kerckhoffs Institute. Simply send an email to [info@kerckhoffs-institute.org](mailto:info@kerckhoffs-institute.org) stating your full name, date of birth, student number and the university you registered at.

Finally, you also need to register as a full-time computer science student at the Eindhoven University of Technology and the University of Twente (in order to follow the courses offered by those universities). These secondary registrations are free of charge. Make sure you include a "Bewijs van Betaald Collegegeld" (proof of admission fee paid) issued by the Central Student Administration (CSA) with the registration request.

Students with a Bachelor degree in Computer Science from a Dutch university are automatically admitted to the security master program. Registration is of course still required.

Students from a polytechnic (in Dutch: HBO) with a degree in computer science will also be admitted, but need to follow extended program with additional courses that aim to cover deficiencies in their background.

Foreign students, or students with a different academic degree will be admitted on a case by case basis. A personal interview may be part of the admission procedure.

For more information, contact [info@kerckhoffs-institute.org](mailto:info@kerckhoffs-institute.org) or Peter van Rossum ([petervr@cs.ru.nl](mailto:petervr@cs.ru.nl)). See also the website at <http://www.kerckhoffs-institute.org>.

## 2.4 Subsidiary Subject

In the Management variant and in the Communication variant students choose a subsidiary subject in management (25 ec) resp. a subsidiary subject in communication (21 ec). See the descriptions of the variants.

In the Research variant the subsidiary subject (25 ec) has to be chosen from the master courses offered by the Radboud University. For example you could choose a subsidiary subject in Math, Astrology, Psychology, Art or Artificial Intelligence. It is also possible to do your subsidiary subject at another university, or even abroad.

The subsidiary subject needs written approval of the examination committee. You may send an email to: [examencommissie@cs.ru.nl](mailto:examencommissie@cs.ru.nl)

Examples (partly in Dutch):

### *De menselijke kant van Mens-Machine-interactie*

BKI230 Neurale netwerkmodellen (8 ec)

MKI32 Gebruikersmodellering en -ondersteuning (6 ec)

MKI30 Verdieping human-computer interactie (6 ec)

NBO54B Inleiding Patroonherkenning (6 ec)

### *Security in Society*

Information Law (7 ec) or Knowledge and entrepreneurship (3 ec)

Dynamics of law (7 ec)

Criminology (7 ec)

Introduction to Language and Speech Technology (10 ec)

### *Social Psychology*

Attitudes (6 ec)

Sociale Cognitie (6ec)

Sociale Psychologie (3 ec)

Reclame Psychologie (6 ec)

Sociale beïnvloeding in de praktijk (6 ec)

## 2.5 Final thesis

### **Master's thesis**

On the site of the Master Thesis Lab (<http://www.cs.ru.nl/mtl/>) students find all the guidelines needed:

- General information
- Procedures
- Projects
- Results

The MTL is a semi-autonomous student research institute. The projects within this institute are planned and carried out by students, and reflect their own fields of interest. At the same time,

the results of these projects serve the purpose of Master Thesis for the final assignment of the studies in Computing and Information Sciences. The MTL research covers purely academic as well as business-oriented goals, and is embedded within larger research projects of the university

Contact the master thesis coordinator Patrick van Bommel ([P.vanBommel@cs.ru.nl](mailto:P.vanBommel@cs.ru.nl)) if you need more information

## **3 Course descriptions**

More detailed information about courses is available on the website of the institute:  
<http://www.cs.ru.nl/iii/onderwijs/opleidingen/colleges1.html>

On this site courses are sorted in alphabetical order. You may click the name of the course for a link to more details.

### **3.1 Course descriptions of the Computing Science courses**

In the following paragraph, you will find course descriptions of the Computing Science courses in alphabetical order.

## Analysis of Embedded Systems

Course id: **100154** 6 ec

spring semester

prof. dr. F.W. Vaandrager

### Website

<http://www.cs.ru.nl/~fvaan/PV/>

### Teaching methods

lectures, projects, individual guidance, tutorial

### Pre-requisites

Familiarity with propositional and predicate logic, finite automata, basic complexity theory, and basic (graph) algorithms is assumed. For instance, you should know what a tautology is, how to formally prove a formula in predicate logic, how to determinize a finite automaton, what is the time complexity of sorting, and how to find the strongly connected components of a graph. The mathematics and theory courses from our bachelor curriculum will certainly provide enough background (frequently, I will refer to topics that have been previously addressed during the bachelor courses such as Discrete Wiskunde, Talen, Beweren en Bewijzen, Inleiding in de Complexiteitstheorie, Logica). If you are not familiar with the concepts, please see the instructor.

### Learning outcomes

Please consult the website

### Description

As our daily lives depend increasingly on digital systems, the reliability of these systems becomes a concern of overwhelming importance, and as the complexity of the systems grows, their reliability can no longer be sufficiently controlled by the traditional approaches of testing and simulation. It becomes essential to build mathematical models of these systems, and to use (algorithmic) analysis methods to explore these models. During recent years there has been enormous progress in the areas of hardware and software verification and analysis. In this course an overview will be presented of mathematical techniques for the specification and analysis of embedded systems. The application of these techniques will be illustrated on industrial sized examples taken from the areas of embedded real-time systems, distributed algorithms and protocols. Participants learn how to make formal models, and how to analyze them using state-of-the-art techniques and tools.

### Subjects

Please consult the website.

### Literature

Offhand-outs, sheets, and recent papers from the literature.

### Examination

This course has "integrated examination". More on the site.

## Image Processing

Course id: **100015** 6 ec

spring semester

dr. T.E. Schouten

### Website

<http://www.cs.ru.nl/~ths>

### Teaching methods

lectures, projects, individual guidance, tutorial, home study

### Pre-requisites

Admission to the master Computing Science at RU is a sufficient basis. Other students are requested to contact the lecturer to check whether their qualifications especially for the assignments (mainly programming in C and C++) are sufficient.

### Learning outcomes

You are able to develop plans to solve image processing problems, taking all steps in the processing chain into account, including interaction with other equipment or systems and with a human operator.

You are able to evaluate program packages, toolboxes, algorithm and methods in relation to their usefulness to solve a specific application.

You are able to further specify, implement and test image processing algorithms with are described in e.g. scientific publications.

### Description

Images are important for receiving information from our environments. They are used in many application areas, e.g. medical diagnosis, remote sensing of the earth, product quality control, security cameras, etc. You will learn how to solve image processing problems developing and using algorithms and methods.

### Subjects

See web pagina for the topics.

### Literature

R.C. Gonzalez & R.E. Woods, Digital Image Processing, Second Edition, Prentice-Hall, ISBN 0-201-18075-8.

Students need to have a sufficient access to this book for work on the assignments.

### Examination

A sufficient participation in the lectures and presentations is required. In case too many lectures or presentations are not attended for valid reasons, this can be replaced by an additional assignment.

The grade of the course is determined by the (weighted) average of the grades for each assignment.

Details like maximum number of lectures to miss, number of assignments and their weights, are presented at the first lecture.

## Business rules

Course id: **IMK001** *6 ec*

fall semester

prof. dr. H.A. Proper

### Learning outcomes

After attending this course, students are able to:

- argue about the role of business rules in enterprises and society,
- compare and position different strategies/approaches for the formulation and rationalisation of business rules,
- compare and position different languages to represent business rules,
- compare and position different strategies/approaches for the deployment of business rules.

### Description

The behaviour of modern day enterprises, as well as society, are largely determined *rules*. Examples of such rules are:

1. Taxation laws.
2. Rules governing the application of mortgages.
3. Guidelines guiding doctors in diagnosing patients.

Sometimes these rules reflect *laws* which one would like to enforce strictly. At other times, they represent best-practices that aim to guide people in performing their work.

Collectively one may refer to these rules as *business rules*. Business rules constrain/guide the behaviour of businesses/enterprises/, both with regard to operational processes as well as change processes. In this new course we will investigate several aspects of such rules. For example, the modelling processes required to obtain these rules, the languages needed to express the rules, as well as the measurements needed to enforce them. With regards to the process of modelling business rules, we will take the perspective that this involves a specific kind of knowledge engineering since business rules essentially capture organisational knowledge.

Finally, business rules are a means to an end. Some parties must have some reason to regulate the behaviour of/in an enterprise/society. In this need we find the rationalization for business rules and their deployment. We will take a risk management perspective in reasoning about such regulatory need, allowing us to take a cost-benefit perspective on business rules and their deployment.

### Subjects

- Business rules,
- Domain modelling,
- Knowledge engineering,
- Regulation of systems,
- Risk management

### Examination

50% A written exam with questions pertaining to the papers read during the course.

40% The case study.

10% Presentation(s) of read paper(s).

## Cognition and Representation

Course id: **100054** 6 ec

fall semester

dr. J.J. Sarbo

### Website

<http://www.cs.ru.nl/~janos/CR.html>

### Teaching methods

lectures, individual guidance, tutorial, home study

### Pre-requisites

propositional logic

### Learning outcomes

- In computer science, in particular, the term 'representation' corresponds to formalisation. We learn how such formalized knowledge can arise from knowledge that is experienced.
- In this course we focus on the second understanding of knowledge. We try to answer the question: What is there in the 'real' world, and how can it be systematically specified?
- We introduce a formal method which based on a theory of signs (semiotics), and learn how that method can be used for modeling knowledge in various domains like 'naive' logic, 'naive' reasoning), and natural language ((morpho-)-syntax, and semantics).

### Description

In computer science, the term 'representation' corresponds to formalisation, and we learn how formalized knowledge be generated from knowledge that we already have. In cognitive theory, however, the term 'knowledge' or 'cognition' is usually associated with

1. thoughts that are a result of reasoning
2. observations that are obtained through experience

which are intensional hence not formal. In this course we learn (i) how these roughly complementary concepts of knowledge can be linked with one another as well as (ii) with the interpretation of 'knowledge', as a formal computation. In this course the focus is on the process how knowledge arises through the observation of phenomena. An example of such a process is the studying of problems (which too appear as phenomena).

The fundamental question raised by this course is this: How can 'real' world phenomena be specified systematically? To this end we introduce a uniform representation of knowledge on the basis of an analysis of the properties of cognition and the processing of signs. Additionally we learn how such a representation can be used for modeling knowledge in the different domains of knowledge like 'naive' logic, 'naive' reasoning, 'naive' or natural language ((morpho-)-syntax, and semantics) and 'naive' mathematics (the conception of number.

### Subjects

Consult the website.

### Literature

Lecture notes

## Complexity theory

Course id: **I00023** 6 ec

spring semester

dr. D.C. van Leijenhorst

### Website

<http://www.cs.ru.nl/~bolke/CXT/BeschrCXT.html>

### Teaching methods

lectures, individual guidance, home study

### Pre-requisites

Prerequisites: basic courses (II, WD1/2, Wk, Wl, T1, T2, WA, P2); in particular T1 en T2.

### Description

The CXT course is about the complexity of algorithms, that is, the costs: fast programs; division into classes; upper and lower bounds; computation models. It is a sequel to the T2 course (Analysis of Algorithms and Intr. to Complexity).

A large part of this CXT course is classical stuff that every computer scientist is supposed to know. However, some more specialist matters will be treated, since CXT is a 'caput course': various important techniques and insights will be presented by studying a number of nice and important subjects. Let us mention:

### Subjects

- Gaining speed at the expense of memory space: tabular methods, dynamic programming (applications: parsing, matrix multiplications, the unary partition problem);
- What everybody should know about polynomials: straight line programs, optimality of the Horner scheme, fast evaluation and interpolation; applications: e.g. threshold schemes (used to cryptographically distribute responsibilities within an organisation);
- Circuit complexity. Nice and clever constructions of logical circuits. Among others, I shall prove that in order to write down a formula for an  $n$  by  $n$  determinant over  $\{0,1\}$  one needs at least  $Cst. n^3$  sheets of scrap paper... whatever ingenious method you might devise to group terms and place brackets!
- Some combinatorics: Generating permutations and combinations really fast!;
- Algorithmic (Kolmogorow-) complexity. The information content of strings - how far can a text be compressed? What is "randomness"? Is it possible to prove a one kilogram theorem with one pound of axioms? Some elegant applications.

### Literature

There are course notes that can be downloaded. To be found together with more information about CXT at:

<http://www.cs.kun.nl/~bolke/Teaching.html>

### Examination

Examination is in principle orally, or in the form of some small project.

## Computational Intelligence

Course id: **100025** 6 ec

spring semester

dr. P.J.F. Lucas

### Website

<http://www.cs.kun.nl/~peterl/teaching/CI/>

### Teaching methods

lectures, projects, individual guidance, tutorial, home study

### Learning outcomes

- understand the principles of reasoning under uncertainty
- understand different numerical models for the representation of uncertainty, such as the CF model, the subjective Bayesian method, Bayesian belief networks, and possibly Dempster-Shafer theory
- have insight into model-based approaches to AI
- have insight into the pros and cons of learning models versus using expert knowledge
- have some experience in experimenting with computational intelligence systems to solve problems involving probability theory

### Description

Handling uncertain knowledge has been one of the central problems of AI research during the past 30 years. In the 1970s and 1980s uncertainty was handled by means of formalisms that were linked to rule-based representation and reasoning methods. Since the 1990s probabilistic graphical models, in particular Bayesian networks, are seen as the primary formalisms to deal with uncertain knowledge. Both early and new methods for representing uncertainty are studied in the course, where in particular various aspects of Bayesian networks are covered.

### Subjects

Introduction to Computational Intelligence, Early models of uncertainty, Probability theory, Bayesian networks: principles, Markov independence, Reasoning with Bayesian networks, Building Bayesian networks, Learning Bayesian networks

### Literature

- P.J.F. Lucas and L.C. van der Gaag, Principles of Expert Systems, Addison-Wesley, Wokingham, 1991, Chapter 5.
- K.B. Korb and A.E. Nicholson, Bayesian Artificial Intelligence, Chapman & Hall, Boca Raton, 2004.
- R.G. Cowell, A.P. Dawid, S.L. Lauritzen and D.J. Spiegelhalter, Probabilistic Networks and Expert Systems, Springer, New York, 1999.
- F.V. Jensen, Bayesian Networks and Decision Graphs, Springer, New York, 2001.

### Examination

Written exam in addition to practical work.

## Design of Embedded Systems

Course id: **100155** 6 ec

fall semester

dr. J.J.M. Hooman

### Website

<http://www.cs.ru.nl/~hooman/DES/>

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

Basic programming skills using C and some knowledge of object-oriented programming.

### Learning outcomes

Students are able

- to develop small applications on a real-time operating system
- to apply a systematic development method to design a relatively small embedded system
- to reason about important system properties, such as correctness, extendability, maintainability, real-time behaviour, and reliability
- to discuss design decisions
- to evaluate the methods and tools used

### Description

This course addresses the development of embedded real-time systems, from specification to implementation. The emphasis is on hands-on experience of the students by means of exercises on a real-time operating system and a concrete case study with real-time and fault-tolerant aspects. The case study is developed incrementally using a UML-based CASE tool.

### Subjects

There are two main topics:

- Basic real-time scheduling theory, supported by scheduling tools and hands-on experience with a real-time operating system.
- Iterative, incremental development of a concrete real-time embedded system. The system will be modelled by means of UML (Unified Modeling Language) using a suitable tool. Emphasis will be on making the main system qualities (such as extendability, performance, reliability) explicit and realizing them in the design. Depending on the case study, typical aspects of embedded systems such as timing behaviour, scheduling, memory limitations, etc., will be addressed.

### Literature

Relevant literature will be distributed during the course.

### Examination

- A report about the work on the case study and an evaluation of the approach, tools, design, etc.;
- A demo and an oral discussion about the report and the contents of the course.

## Advanced Programming

Course id: **100032** *6 ec*

fall semester

prof. dr. ir. M.J. Plasmeijer  
dr. P.W.M. Koopman

### Teaching methods

lectures, practical, home study

### Pre-requisites

Bachelor Computer Science.

### Learning outcomes

Insight in some state-of-the-art advanced programming concepts, their theoretical background and their practical applicability.

### Description

In this course advanced programming concepts will be taught. We will address (1) generic programming; (2) combinatorial programming; (3) advanced concurrency concepts; (4) advanced data structures.

### Subjects

(1) generic programming: theoretical background, overloading, type constructor classes, kind indexing; practical applications: common generic functions, web-applications, workflow applications; (2) combinatorial programming (monadic programming, parser combinators); (3) advanced concurrency concepts (Software Transactional Memory); (4) advanced data structures (Generalized Algebraic Datatypes).

### Literature

Scientific papers on the topics above which will be handed out during the course.

### Examination

Written, closed book.

### Extra information

The course is part of the theme "Software Construction"

For practical experience we use the state-of-the-art functional language Clean.

## Foundations of Information Systems (IS2)

Course id: **I00035** *6 ec*

spring semester

dr. P. van Bommel

### Teaching methods

lectures, tutorial, home study

### Learning outcomes

We study fundamental aspects of information systems in detail. You will learn: (1) transformation of information models, (2) reasoning about transformations, and (3) optimization of transformations.

### Description

We consider information modelling and transformation. We emphasize transformations in which no semantics is lost (lossless transformations).

### Subjects

We do this by means of:

- Generation operators and mutation operators.
- Application of these operators in automated design processes.
- Prediction of the behaviour of transformation strategies.

### Literature

The lecture notes will be given via blackboard.

### Examination

Student paper and examination.

### Extra information

Keywords: databases, transformation, optimization.

The course consists of hoorcolleges and werkcolleges. The students also have to write a paper.

## ICT & Samenleving 2

Course id: **I00037** 3 ec

spring semester

dr. L. Consoli

### Website

[http://www.ru.nl/fil-beta/lucac/ICTS2\\_2006/icts2\\_index.html](http://www.ru.nl/fil-beta/lucac/ICTS2_2006/icts2_index.html)

### Teaching methods

lectures, project, home study

### Learning outcomes

Aan het einde van de cursus de student:

1. heeft kennis van een aantal ethische en filosofische theorieën, zowel in het algemeen als toegepast op het eigen vakgebied
2. heeft kennis van de filosofische veronderstellingen en implicaties van een aantal cutting edge onderzoeksvelden binnen zijn/haar vakgebieden
3. is zich bewust van de noodzaak van zelfreflectie om de eigen normativiteit ter discussie te stellen.
4. is in staat om die veronderstellingen en implicaties te expliciteren
5. is in staat om zelfstandig onderzoek uit te voeren naar een relevant onderwerp en de resultaten daarvan te presenteren in een schriftelijk betoog

### Description

The course Informatics and Society 2 (Informatica en Samenleving 2) explores some cultural and social issues that have been made possible by the development of information technology. Starting from the concepts of privacy en the relationship between informatics and political decision-making processes, we will move on to examine the changes in our way to look at the world caused by the progresses of information technology. On one hand we will study the social/ethical/cultural implications of artificial intelligence (AI); on the other hand we will put forth as a central theme the feasibility of software in all its implications

### Subjects

Algemene theorievorming: Inleiding in de hoofdstromingen van de ethiek (deontologie, utilisme, deugdethiek); Inleiding de mind-body problem; Inleiding in systeemtheorie

Studiespecifieke onderwerpen

1. Vrijheid van meningsuiting (verhouding met privacy; het World Wide Web; relatie vrijheid van meningsuiting vrijheid van informatieuitwisseling
2. Hacker- ethiek (old vs. new hackers, open source vs. closed source, Free Software Foundation en Richard Stallmann, Linux vs. Windows)
3. Safety-critical systems (inleiding; ethische dilemma's, professionele verantwoordelijkheid)
4. Expert systems (filosofische veronderstellingen, expert systems en ethische dilemma's)
5. Artificial Intelligence (filosofische betekenis van kunstmatige intelligentie, language-based vs. pattern-based benadering, swarm intelligence, robots en cyborgs, implicaties van AI voor de ethiek, artificial intelligence vs. artificial life)

## Information Retrieval

Course id: **100041** 6 ec

spring semester

prof. dr. ir. T.P. van der Weide

### Website

<http://www.cs.ru.nl/iii/onderwijs/opleidingen/colleges1.html>

### Teaching methods

lectures, tutorial, home study

### Pre-requisites

Participant of IR (A constructive approach to Information Retrieval) should have the base qualifications as provided by the bachelor Informatica or Informatiekunde.

### Learning outcomes

The goals of the course IR (A constructive approach to Information Retrieval) is that its participants

1. are familiar with the base models that are used for Information Retrieval.
2. have knowledge of query languages, both syntactically and semantically.
3. are familiar with information extraction from documents, inter-document relations and their appreciation.
4. have insight and proficiency in design and construction of search engines.
5. have insight in interaction techniques to support searchers in their quest for information.
6. have some experience with scientific literature in this field.

### Description

IR (A constructive approach to Information Retrieval) treats the backgrounds of Information Retrieval:

1. How do people search for information, and how can this be formalized?
2. How do people describe what they mean, and how can we formalize meaning?
3. How can these points be combined?

An important application area is the Internet.

### Subjects

The course consists of three main parts:

1. Fundamentals
2. Knowledge extraction and Information processing
3. Document relations on the Web

During the course, guest speakers are invited to discuss state-of-the-art topics.

For more information, please consult the website.

### Literature

Lecture notes will be made available via Blackboard.

### Examination

Please consult the website.

## ICTproject in a different culture

Course id: **100171** 6 ec

fall semester

prof. dr. ir. T.P. van der Weide

### Teaching methods

lectures, projects, individual guidance, tutorial, home study

### Pre-requisites

Het vak ICT & Samenleving 1 wordt ten zeerste aangeraden.

### Learning outcomes

Na afloop van deze cursus kan de student

1. meewerken aan een project gericht op verbetering van de ICT infrastructuur in een andere cultuur;
2. zich efficiënt oriënteren op de bijzondere omstandigheden in de context van het project;
3. de eigen rol definiëren binnen zo'n project; deze rol plaatsen in het kader van eigen persoonlijke of vakinhoudelijke ontwikkeling;
4. reflecteren op de toepasbaarheid van eigen kennis en vaardigheden in de nieuwe context;
5. reflecteren op de eigen bijdrage, voortgang en resultaten.

### Description

De student werkt mee in een lopend community project. Het werk heeft het karakter van een stage: het meewerken staat centraal. Daarnaast is er de voorbereiding en het reflecteren op het eigen handelen.

In het academische jaar 2007-2008 wordt deze stage als pilot aangeboden binnen het Onderwijsinstituut voor Informatica en Informatiekunde. Het is beperkt toegankelijk voor studenten Informatica en Informatiekunde. Na evaluatie en eventuele bijstelling wordt er vanaf 2008-2009 een variant ontwikkeld waaraan ook andere FNWI-studenten kunnen meedoen. De deelname aan deze stage wordt beperkt door de mogelijkheden die binnen de lopende projecten voorhanden zijn.

### Examination

Aan de hand van positieve voortgangsbeslissingen (ter beoordeling aan stagecoördinator en eventuele lokale begeleider) en het complete stageverslag (voorstudie, stagevoorstel, activiteitenverslag, reflectie). Als de student de stage succesvol doorloopt, is de beoordeling 'voldaan'.

## Proof Assistants

Course id: **100139** 6 ec

spring semester

dr. F. Wiedijk

### Website

<http://www.cs.ru.nl/~freek/courses/pa-2007/>

### Teaching methods

lectures, home study

### Pre-requisites

Logic, Semantics, Functional Programming, Basic Mathematics courses.

### Learning outcomes

Understand the general principles behind the implementation of proof assistants and to be able to use them to implement simple decision procedures. Obtain an overview of the different proof assistants that are used, how they differ and what their respective strengths are. Obtain enough experience with at least one of the systems to be able to do a theory and proof development in it.

### Description

The course Proof Assistants treats the implementation and use of proof assistants like Coq, PVS, HOL en Mizar. In the first part, we present the LCF-approach towards implementing proof assistants and show some decision procedures and how they are implemented. In the second part we show how mathematical proofs are formalized in practice.

### Subjects

- Implementation of Proof assistants
- LCF-approach
- Procedural and Declarative proof styles
- Tactics
- Decision procedures (Automated search in first-order logic Reasoning with equality by congruence closures and rewriting techniques)
- Proof objects
- Coq, PVS, HOL, Mizar
- Formalizing Mathematics

### Literature

Selected Chapters from "Introduction to Logic and Automated Theorem Proving" by John Harrison. This book has not yet appeared. If necessary, printed copies will be handed out at the lectures. Course Notes.

### Examination

Each student will do a few small projects. At the end a group presentation has to be given.

## R&D: System Development

Course id: **100156** 8 ec

fall semester

dr. M.C.J.D. van Eekelen

### Website

<http://www.cs.ru.nl/~marko/onderwijs>

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

SE, SO1

### Learning outcomes

Goals- Learn to cooperate in somewhat larger software projects using **innovative** methods, tools or techniques- Acquire theoretical and practical insight in the innovative software development process - Learn to function as senior software engineer - Recognise and apply concepts and models for software development

### Description

Set up: - Simulation of actual development of innovative software in projects- Assisted by some lectures

### Literature

Roger S. Pressman, Software Engineering, A Practitioner's Approach - European Adaptation, SIXTH Edition, McGraw-Hill, ISBN 0-07-301933-X

### Examination

projectevaluations, projectreport, exam on the literature

## **R&D: System Development Management 1**

Course id: **100157** 6 ec

spring semester

dr. T.E. Schouten

### **Website**

<http://www.cs.ru.nl/~ths>

### **Teaching methods**

lectures, projects, home study

### **Pre-requisites**

Bachelor Computer Science

### **Learning outcomes**

SDM1 has the aim that the student, at the end of the course, has all the professional skills of an IT project leader.

### **Description**

SDM1 resembles the phase in an IT career in which the project leader takes responsibility for the management of a software development project. Within SDM1 we address the project management aspects of the whole life cycle of a system development project, from definition study through system design, system development and system implementation all the way to the maintenance of a system in an operational environment.

The course consists of a theoretical and a (large) practical component. The practical component is being carried out within "GiP-House", managing students from the "Software Engineering course". GiP-House closely resembles a real-life modern softwarehouse.

### **Subjects**

See the website.

### **Literature**

Software Engineering A practitioners Approach: European Adaptation, sixth edition, Roger S. Pressman

Sheets of the presentations

### **Examination**

- Serious participation in the practical part is required. If this is judged insufficient the student has failed the course, and is not allowed to take part in the written examination
- a 2 hour written examination, this determines the final grade. No literature may be consulted during the examination.

## R&D: System Development Management 2

Course id: **100158** 6 ec

fall semester

prof. dr. M. van Vliet

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

1. A student has experience in one of the manager roles as executed in SDM1
2. A student is capable to work in a real-life project situation at the level of a *senior software engineer*.
3. A student is able to work within teams (GiP-House).
4. A student has enough knowledge of methods and tools to execute innovative software engineering projects.

### Learning outcomes

The course consists of a theoretical and a (large) practical component. The practical component is being carried out within 'GiP-House'. GiP-House closely resembles a real-life modern softwarehouse.

SDM2 has the aim that the student, at the end of the course, has all the professional skills of a manager of a software-house. SDM2 resembles the phase in an IT career in which the manager takes responsibility for the management of a software-house. Within SDM2 the student is involved in the management of the students that follow the SDM2 course and directs the projects that the GiP-House student are involved in (tactical management). Furthermore, the SDM2 student is involved in preparing the GiP-house for the future (strategic management). This involves acquisition of new projects, the structure and management of the GiP-House, internal improvement projects within GiP-House and implementing new system development methodologies within the projects of GiP-House.

Within SDM2 we expect from the student an academic working habit, in which taking responsibility for ones' own actions and the corresponding achieved results, is a very important aspect.

For further information with regard to the GiP-House, please contact [ths@cs.ru.nl](mailto:ths@cs.ru.nl)

### Description

The course 'System Development Management 2 (SDM2)' is part of the System Development Management (SDM) stream. SDM2 is a follow-up course on SDM1. SDM2 has the aim that the student, at the end of the course, has all the professional skills of a manager of a software-house. The development of the student as a manager depends on the chosen career-path by the student. At the end of the course this is evaluated in a Professional Role Review.

### Subjects

Within SDM2 the student is, as a manager, responsible for the software-house environment in which systems are developed based on a realistic case. Hereby, the following tasks are assigned during SDM2. At the end of SDM2 the student should have build up competencies in these areas:

1. To work as a manager in a team with clearly defined tasks, deliverables and constraints.
2. To work at an academic level: to be able to assess ones' own performace an the ability to improve upon that.
3. To work independently with regard to software development and to assess and improve the quality of the process.
4. To guide, asses and implement the most suited software development methodologies for the given case.
5. To be able to guide the future development of a software-house.
6. To select and implement the most appropriate software development tools for the given case.
7. To select and implement the best system-architecture (programming languages, software development tools, hardware, middleware, and communication components) for the given case.
8. To be able to guide the process of building, testing and implementing a system with the aim to implement a reliable, maintainable, well-documented and well-tested system (quality assurance).
9. To be able to recognize communicative and organizational issues in a software-house environment and to effectively act upon that.
10. To be able to take account of the human, organizational or social consequences (possibilities, constraints, boundaries and risks) of the management of a software-house in the real-life world.
11. To be able to convey knowledge in both oral as in written form. To be able to present and defend a proposal and to be able to be effective in written and oral communication in a project environment.
12. To be able to assess ones' own performace and to report upn that.
13. To be able to work independently on ones' own professionalism: To gain and create knowledge based upon own literature survey and gained advice.

### **Literature**

1. Course books form the GiP-House library.
2. Course material handed out during the course.
3. Online GiP-House handboek.(<http://www.cs.kun.nl/is/edu/gip/hb/inhoudsopgave.html>).
4. GiP-House website (<http://www.giphouse.nl>).

### **Examination**

It is necessary that a student works seriously within the practical part of SDM2. Without serious participation in the practical component, a student shall not receive grading for this course. At the end of the course there is a written exam on the subject lectured in the theoretical component of SDM2.

## R&D: System Development Research

Course id: **100162** 6 ec

fall semester

dr. M.C.J.D. van Eekelen

### Website

<http://www.cs.ru.nl/~marko/onderwijs>

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

SE, SO1, Research Lab 1,2

### Learning outcomes

Doelstelling Goals- Learn to cooperate in somewhat larger software projects using **innovative** methods, tools or techniques in a research environment- Acquire theoretical and practical insight in the innovative software development process in a research environment- Learn to function as senior research software engineer- Recognise and apply concepts and models for software development in a research environment

### Description

Set up: - Simulation of actual development of innovative software in projects in an actual research environment- Assisted by some lectures

### Literature

Roger S. Pressman, Software Engineering, A Practitioner's Approach - European Adaptation, SIXTH Edition, McGraw-Hill, ISBN 0-07-301933-X

### Examination

on the basis of report and projectevaluation combined with an exam on the literature

### Extra information

Project proposals have to be done by the students after consulting a research group. This can continue on a project in a previous research lab course or it can be preparatory for a Master thesis subject or an extension of a Master thesis subject. Furthermore, specific proposals connected to ongoing research in a research group can be done.  
project, report, draft publication

## **R&D: Research 1**

Course id: **100159** 8 ec

fall semester

dr. E.M.G.M. Hubbers

### **Website**

<http://www.cs.ru.nl/~hubbers/courses/rdr1>

### **Teaching methods**

lectures, projects, individual guidance, home study

### **Learning outcomes**

After the course the student should be able to do scientific research. He should be able to deal with the following topics:

- Choosing a subject
- Formulating a proper research question
- Writing a complete research proposal
- Performing literature study
- Combining the results found into a readable paper
- Reviewing other papers

### **Description**

For this course students have to set up and complete a small research project.

### **Subjects**

Usually there is a mandatory general topic. Within this framework students are free to choose their own favourite topic.

### **Literature**

Students have to find their own literature.

### **Examination**

The final grade is mostly based upon the paper written by the student. However, the smaller assignments like writing a referee report also contribute to the final grade.

## R&D: Research 2

Course id: **100160** 6 ec

spring semester

dr. E.M.G.M. Hubbers

### Website

<http://www.cs.ru.nl/~hubbers/courses/rdr2>

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

You must have finished RDR1 successfully.

### Learning outcomes

After the course the student should be able to come up with a good reasearch proposal and act on that proposal. Important aspects are:

- Selecting appropriate scientific methods for answering the research question.
- Working together with an expert in the research area.
- Being able to put own results in the scientific context.
- Transforming technical results into readable text.
- Presenting the research.

### Description

This course builds upon the knowledge acquired within R&D: Research 1 (RDR1). However, in RDR2 the focus is shifted from a literature study to adding really new contributions to the research area. Obviously, the results have to be presented in a paper and in a presentation.

### Subjects

Each student chooses his own supervisor within the department. Together they come up with an appropriate research question.

### Literature

Students have to find their own literature.

### Examination

Main input for your final grade are the comments from your supervisor. Since he is the expert he knows what your article is worth. He also knows how you did the research (much guidance needed or not). This grade can be adjusted based upon a final oral exam in which not only your own research but also that of others can be discussed.

## **R&D: Research 3**

Course id: **100161** 6 ec

fall semester

dr. E.M.G.M. Hubbers

### **Website**

<http://www.cs.ru.nl/~hubbers/courses/rdr3>

### **Teaching methods**

lectures, projects, individual guidance, home study

### **Pre-requisites**

Completed RDR2 successfully.

### **Learning outcomes**

R-track: develop own skills of doing research. S-track: learn how to manage a research project. Identify pitfalls in the research process of someone else.

### **Description**

The course R&D: Research 3 (RDR3) includes two tracks: a research-track (R-track) and a supervisor-track (S-track). The R-track is basically the same as RDR2: you have to do research and describe your results in a paper. The S-track means that you are going to supervise some RDR1 students with their project. Apart from this supervision task one also has to study some papers and write a referee report about them. Furthermore, an additional task will be introduced to the S-track. Most likely this will imply that you have to write a 'popular' article about a research topic for a newspaper.

### **Subjects**

In th R-track students choose their own topic. In the S-track the RDR1 students choose the topic.

### **Literature**

Students have to find their own literature.

### **Examination**

For the R-track the quality of the paper and presentation imply the final grade. For the S-track the quality of the evaluation report, referee reports and the additional task will reflect in the final grade.

### **Extra information**

Depending on your results of RDR2, personal interest and the number of students in RDR1 and RDR3, the teacher will decide which track you are going to follow. For both tracks there will be some progress meetings.

## Reliability of Software Systems

Course id: **I00018** 6 ec fall semester

dr. M.C.J.D. van Eekelen  
prof. dr. ir. M.J. Plasmeijer

### Website

<http://www.cs.ru.nl/~marko/onderwijs/bss>

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

Bachelor

### Learning outcomes

Experience with tools for assessing reliability aspects of software systems  
Experience with level of reliability in practice  
Experience with practical applicability of academic tools  
Academic evaluation of applicability of reliability tools

### Description

Reliability of Software Systems:

Case studies in software reliability, preferably in the context of LaQuSo, the Laboratory for Quality Software of TU Eindhoven and RU Nijmegen.

### Subjects

Many different subjects: varying from security issues via static analysis, to formal verification, testing and performance analysis. Tools that can be used are e.g. JML, Spec#, Findbugs, Esc-Java, Sparkle, PVS, Yasper, mcr12, Gast, Torx.

### Examination

Evaluation Report, End presentation

### Extra information

A few lectures for general instruction followed by a case study guided by a supervisor. Regular supervisory meetings. Regular progress meetings with all case study groups.

## Security in organisations

Course id: **100153** 6 ec

fall semester

drs. P.J.B. van Rossum

### Website

<http://www.cs.ru.nl/~martijno/secorg/>

### Teaching methods

lectures, projects, individual guidance, home study

### Pre-requisites

The bachelor security course Security.

### Learning outcomes

- To develop a suitable level of paranoia, needed for designing and deploying security sensitive IT applications
- To learn how to manage risk while designing and deploying IT systems within an organisation
- To learn how to write and enforce good security policies
- To learn some basic techniques for evaluating security solutions

### Description

Security is about regulating access to assets. Crucial questions are: Who are you? and: Should you be doing that? These questions cannot be answered without taking the organisation in which security systems are deployed into account.

The course introduces the basic notions and techniques in the area of information security. The emphasis lies on assessing risks and impacts of security related threats and on planning, managing and evaluating security controls such as policies, procedures, and technical counter measures.

### Subjects

Security in context; Assets and threats; Risk, vulnerability, control, attack, damage; Risk assessment and risk management; Methods/tools for risk analysis; Attack trees; Security policies; Roles, Classifications; Code of Practice for Information Security (BS7799); Evaluation of Information Security; Business continuity planning and incident recovery CERTs; Legal Aspects

### Literature

Jan Killmeyer Tudor -Information Security Architecture:An Integrated Approach to Security in the Organization -Auerbach publishers (1st or 2nd edition)

### Examination

Mandatory assignments and mandatory written exam.

## Security seminar

Course id: **100136** 6 ec                      spring semester

### Website

<http://www.cs.ru.nl/~jhh/secsem.html>

### Learning outcomes

Student knows the state of the art in privacy enhancing technologies, and is aware of their limitations when applied in practice.

Student can efficiently give a lecture on a selected topic, and can write a readable and technically sound paper on another selected topic.

### Description

Privacy has always been a controversial topic. Governments and business want to collect information about their citizens and customers - for their own benefit as well their clients. In surveys, people claim that they value their privacy. In practice, people give away personal information very easily, either because they do not receive a service otherwise, or because they are unable to protect that information reliably. With the calls for ever increasing security - after the events of the last few years - privacy has eroded even further, it seems. In this seminar we will explore the state of the art in privacy enhancing technologies (PET), and discuss theories (technical, legal and societal) of privacy.

### Subjects

Privacy, anonymity, mixnets, voting, legal and societal theories of privacy.

### Examination

Writing a final paper.

### Extra information

To successfully complete this course the student has to present a topic of his choice in class, write a final paper on another topic of his choice, and review a final paper of one other student. Seminar.

## Semantics

Course id: **100088** 6 ec

fall semester

prof. dr. J.H. Geuvers

### Website

<http://www.cs.kun.nl/~wil>

### Teaching methods

lectures, tutorial, home study

### Pre-requisites

Basic knowledges of operational semantics of functional languages and of imperative languages (T3 or Semantics and Logic 1 and 2). A good knowledge of the topic "lambda-calculus" of these courses: beta-reduction and  $\lambda$ -conversion, fixed points, Church-Rosser property, representation of recursive functions in the lambda-calculus.

### Learning outcomes

Knowledge of cpos, reduction behavior of lambda terms, interpretation of terms in models, operational and denotational semantics of imperative programs

### Description

Semantics of functional and imperative languages

This course is an important theoretical foundation for the courses on functional languages, courses in programming language design and in general for courses in which language design and compiling plays an important role. There is a strong connection with the course Type Theory.

### Subjects

Reduction strategies for lambda terms (operational semantics for functional languages), interpretation of lambda terms in models (denotational semantics for functional languages), continuous functions on complete lattices and complete partial orderings (cpo's), operational and denotational semantics of imperative programs with recursive procedures, validity of correctness formulas.

### Literature

Hanne Riis Nielson and Flemming Nielson: Semantics with applications, 1999 (via internet: PDF)

2Course notes will be handed out at the lectures.

### Examination

Written exam

## Software Security

Course id: **ISOFSE** 6 ec

spring semester

dr. ir. E. Poll

### Website

<http://www.cs.ru.nl/~erikpoll/SoftwareSecurity>

### Teaching methods

lectures, practical, projects, individual guidance, tutorial, home study

### Pre-requisites

Basic programming skills, in particular knowledge of Java (enough to write a non-trivial application for the project work) and basic knowledge about C/C++.

### Learning outcomes

- students are able to identify security objectives of software applications;
- students understand typical ways in which software fails to be secure;
- students know methods and technologies that can help in the development of secure software, and are able to select and apply these.

### Description

Bad software is an important - if not *the* most important - cause of security problems. This course is about the challenges in developing secure software and the technologies that can be used to improve software security.

### Subjects

What is software security?; Common software vulnerabilities: lack of input validation (buffer overflows, SQL injections, etc.), race conditions, broken access control, etc. Design flaws. Implementation flaws. Deployment flaws. Case studies; Guiding principles; Architecture. Design. Implementation; Access control; Language level security: typing; tainting input data; untrusted code security; Application level security: runtime monitoring; static analysis; verification; proof-carrying code; JML, Spec#, Software evaluation; Case studies.

### Literature

Selected technical papers that are publicly available on the web.

Interesting background material to read are the books

- Building Secure Software, by John Viega and Gary McGraw. Addison-Wesley, 2002.
- Secure Coding: Principles & Practices, by Mark G. Graff and Kenneth R. van Wyk. O'Reilly, 2003.
- The 19 Deadly Sins of Software Security, by Michael Howard, David LeBlanc and John Viega, McGraw-Hill, 2006.

both of which are available in the library.

### Examination

Project assignments.

## Testing techniques

Course id: **100110** 6 ec

fall semester

dr. ir. G.J. Tretmans

### Teaching methods

lectures, projects, individual guidance, tutorial, home study

### Pre-requisites

Some programming experience; Some experience with formal methods in software development.

### Learning outcomes

The goals of the course "testing techniques" are that students: (1) obtain an overview of, and insight in the importance and the place of structured and systematic testing within the software development cycle; (2) are familiar with standard testing concepts, terms, and nomenclature; (3) know and recognize different kinds, phases, and aspects of structured testing; (4) develop skills in applying some techniques for developing tests; (5) get experience in applying some test tools; (6) have knowledge of, and can apply some of the latest research results in model-based testing.

### Description

Testing is part of almost any software development project. The course "testing techniques" deals with a number of techniques, methods and tools which may help in the systematic and effective testing of software systems. Established testing techniques as well as some new developments, such as model-based testing, will be presented. Some guest lectures will be given by people from industrial software testing.

### Subjects

Topics which will be discussed are: kinds, aspects, and phases of software testing, testing within the software development trajectory, test automation and test tools, (code-)coverage, test organization, testing standards, developing test cases (black-box, white-box, equivalence partitioning, boundary value analysis, state-based testing), automatic test generation, the discriminating power of tests (testing equivalences), model-based testing with transition systems.

### Literature

For part 1: D. Graham, E. van Veenendaal, I. Evans, R. Black: Foundations of Software Testing: ISTQB Certification. Thomson Learning, 2007;  
<http://www.thomsonlearning.co.uk/istqb/>. For Part 2: Reader (via Blackboard)

### Examination

Homework assignments; Examination

## Type theory

Course id: **100111** 6 ec

H

dr. F. Wiedijk  
dr. M. Niqui

### Website

<http://www.cs.ru.nl/~milad/courses/TT2007/>

### Teaching methods

lectures, tutorial, home study

### Pre-requisites

\* Je dient over enige basiskennis van de (ongetypeerde) lambda-calculus te beschikken. Deze kennis is aanwezig na het volgen van het vak T3: Semantiek en correctheid \* Daarnaast is enige ervaring met propositie- en predicatenlogica van belang. \* Voor het doorzien van het verband tussen theorie en praktijk komt ervaring met zowel imperatieve als functionele programmeertalen van pas. Die ervaring is aanwezig na het volgen van de Programmeerlijn in de basisopleiding. \* Bekendheid met constructieve of intuïtionistische wiskunde (bijvoorbeeld via het vak Intuïtionistische wiskunde van W.H.M. Veldman) is voor de bewijsvariant zeer nuttig maar niet noodzakelijk (zie verderop). \* Enige discipline voor het werken met formalismen is gewenst. Doorgaans is die aanwezig na het volgen van de basisopleiding.

### Learning outcomes

Please consult the website (in Dutch)

### Description

Lambdatermen hebben nu types, net als (delen van) programma's in programmeertalen. Getypeerde lambda calculi zijn nauw verbonden met functionele talen. Bovendien is er een sterk verband met logica. Dit geeft aanleiding tot 'proof checkers'. We bestuderen en vergelijken verschillende systemen van getypeerde lambda calculi.

### Subjects

Getypeerde lambda-calculi a la Curry en a la Church. Simpele type theorie, polymorfe type theorie, dependent type theorie. Typeringsalgoritmen. Sterke normalisatie. Representatie van logica's en datatypes in getypeerde lambda calculus. Logical Framework. Hogere orde logica Record Typen Inductieve typen Pure Type systems. Interactieve bewijsconstructie met Coq

### Literature

Collegaantekeningen worden verspreid op het college en via de website.

### Examination

Schriftelijk tentamen en een schriftelijke opdracht

## Wiskunde (schakelcursus)

Course id: **100170** 6 ec

fall semester

dr. D.C. van Leijenhorst

### Website

<http://www.cs.ru.nl/~bolke/SchakWis.html>

### Pre-requisites

Instromerschap, basiskennis algebra en analyse.

### Learning outcomes

Het doel van de cursus is operationeel ("kunnen werken met"), maar waar nodig en mogelijk zal gepoogd worden enig dieper inzicht en breder overzicht te verstrekken.

### Description

Er wordt een korte inleiding gegeven in enkele wiskundige onderwerpen die van belang zijn voor de studie informatica. Dit ter aanvulling van deficiënties in de vooropleiding, of ook het ophalen en uitbreiden van ver weggezakte stof.

### Subjects

Enkele stukken elementaire getaltheorie, complexe getallen, calculus en lineaire algebra.

### Literature

Wordt verstrekt.

### Examination

Schriftelijk, onder voorbehoud.

Er wordt nagedacht over een andere onderwijs-en tentamenvorm. Als dit doorgaat krijg je tijdig informatie.

### 3.2 Courses MT-variant

For the course descriptions please consult the website

[http://www.studiegids.science.ru.nl/2007/en/science/prospectus/afstudeervariant\\_management\\_en\\_toepassing/courses/](http://www.studiegids.science.ru.nl/2007/en/science/prospectus/afstudeervariant_management_en_toepassing/courses/)

<b>Business and Society</b> <b>FMT001B</b> 5 ec	fall semester	dr. G.A.N. Vissers prof. dr. B. Dankbaar
<b>Organization Theory</b> <b>FMT002B</b> 5 ec	spring semester	prof. dr. B. Dankbaar
<b>Innovation Management</b> <b>FMT003B</b> 5 ec	fall semester	drs. ing. P.M. Vos ir. L.J. Lekkerkerk
<b>Strategy and Marketing</b> <b>FMT004B</b> 5 ec	fall semester	
<b>Finance and Accounting</b> <b>FMT005B</b> 5 ec	spring semester	drs. R.A. Minnaar

### 3.3 Courses C-variant

For the course descriptions please consult the website:

[http://www.studiegids.science.ru.nl/2007/en/science/prospectus/afstudeervariant\\_communicatie/courses/](http://www.studiegids.science.ru.nl/2007/en/science/prospectus/afstudeervariant_communicatie/courses/)

<b>Framing Knowledges</b> <b>FC0010C</b> 3 ec	first quarter	dr. J.G. van den Born
<b>Knowledge Society</b> <b>FC0011C</b> 3 ec	third quarter	dr. A.F.M.M. Souren
<b>Science &amp; Media: strategies and trends</b> <b>FC0013C</b> 3 ec	second quarter	H.M. Dresen drs. R.P.M.M. Welters
<b>Introduction Science Communication</b> <b>FC001B</b> 3 ec	first quarter	dr. J.G. van den Born
<b>Science and Social interaction</b> <b>FC002B</b> 3 ec	third quarter	dr. J.G. van den Born
<b>Risk Communication</b> <b>FC003B</b> 3 ec	second quarter	dr. A.F.M.M. Souren
<b>Boundary work: the tensity between diversity and sustainability</b> <b>FC0041C</b> 3 ec	fourth quarter	prof. dr. F.W.J. Keulartz drs. I.E.M. Dankelman

## 4 Practical Information

### 4.1 Important names and addresses

#### Faculty of Sciences

Toernooiveld 1, 6525 ED Nijmegen

Huygens building

tel.: 024-3616161 (Radboud University - general phone nr.)

#### Education Bureau for Computing and Information Sciences

*General:*

- Secretary's office: HG02.609, Huygens building  
tel.: 024-3652084

*Staff:*

- *Ms. Resi Westerman, MA*, pr/secretary; R.Westerman@cs.ru.nl
- *Ms. Doris Meier, MA*, pr/secretary; D.Meier@cs.ru.nl
- *Ms. Vera Kamphuis, MA*, head, coordinator of studies of Information Science; V.Kamphuis@cs.ru.nl
- *Ms. Yella Kleijnen*, coordinator of studies of Computing Science; Y.Kleijnen@cs.ru.nl

#### Student advisor for Master students

- *dr. Theo Schouten*, T.Schouten@cs.ru.nl

#### Student advisor for "HBO-instromers" (post-Polytechnic bachelor students)

- Dr. Hanno Wupper, H.Wupper@cs.ru.nl

#### Master's thesis coordinator

- *Dr. Patrick van Bommel*, P.vanBommel@cs.ru.nl  
website: www.cs.ru.nl/mtl/

#### Education Board

- *Dr. Erik Barendsen*, director; E.Barendsen@cs.ru.nl
- *Dr. Sjaak Smetsers*, coordinator master programme Computing Science; S.Smetsers@cs.ru.nl
- *Prof.dr. Erik Proper*, coordinator master programme Information Science; E.Proper@cs.ru.nl
- *Freek van den Berg*, studentassessor; FvandenBerg@student.ru.nl
- *Ms. Vera Kamphuis, MA*, head of the education bureau; V.Kamphuis@cs.ru.nl
- *Ms. Yella Kleijnen*, secretary, Y.Kleijnen@cs.ru.nl

**Education Committee of Computing Science and Information Sciences**

Members of this committee are 4 students in computing science, 4 students in information science and 4 lecturers.

- *drs. Ger Paulussen*, chairman; G.Paulussen@cs.ru.nl
- *ms. Yella Kleijnen*; secretary; Y.Kleijnen@cs.ru.nl

**Examination committee**

- *dr. Marko van Eekelen*, chairman; M.vanEekelen@cs.ru.nl
- *dr. Dick van Leijenhorst*, vice-chairman; D.vanLeijenhorst@cs.ru.nl
- *dr. Theo Schouten*, secretary; T.Schouten@cs.ru.nl

**Coordinator of international affairs for Computing and Information sciences**

- *dr. Janos Sarbo*, J.Sarbo@cs.ru.nl

**Office of administration and exams for students (FSA)**

- Ms. Clementine Hendriks, Ms. Yvonne Mulder,  
opening hours: Monday to Thursday: 13-16 hrs, Friday: 9-12 hrs  
room: HG00.134, Huygens building  
tel.: 024-3652247/024-3653392

**Student affairs office**

- Comeniuslaan 4, Nijmegen  
tel.: 024-3612345  
webpage: [www.ru.nl/students/](http://www.ru.nl/students/)  
See the 'Vademecum' for more information.

**Students' association Thalia (for students of Computing Science or Information Science)**

- [info@thalia.nu](mailto:info@thalia.nu) (general info)  
website: [www.thalia.nu/](http://www.thalia.nu/)

**Alumni association Ninja (for Computing and Information Sciences)**

- Contact: dr. Dick van Leijenhorst, D.vanLeijenhorst@cs.ru.nl  
website: [www.cs.ru.nl/ninja/](http://www.cs.ru.nl/ninja/)

## 4.2 List of lecturers

Bommel, Dr. P. van	52696	HG02.611
Born, Dr. J.G. van den	52269	HG02.814
Consoli, Dr. L.	53065	HG02.824
Dankbaar, Prof. dr. B.	52681	HG02.809
Dankelman, Drs. I.E.M.	52150	HG02.827
Dresen, H.M.	52188	HG02.831
Eekelen, Dr. M.C.J.D. van	53410	HG02.074
Geuvers, Prof. dr. J.H.	52603	HG02.526
Hooman, Dr. J.J.M.	52590	HG02.514
Hubbers, Dr. E.M.G.M.	52713	HG02.061
Keulartz, Prof. dr. F.W.J.	52851	HG02.823
Koopman, Dr. P.W.M.	52483	HG02.049
Leijenhorst, Dr. D.C. van	52232	HG02.541
Lekkerkerk, Ir. L.J.	11931	TvA1 01.35
Lucas, Dr. P.J.F.	52611	HG02.614
Minnaar, Drs. R.A.	11765	TvA1 02.24
Niqui, Dr. M.	52610	HG02.540
Plasmeijer, Prof. dr. ir. M.J.	52644	HG02.047
Poll, Dr. ir. E.	52710	HG02.073
Proper, Prof. dr. H.A.	52613	HG02.622
Rossum, Drs. P.J.B. van	52077	HG02.071
Sarbo, Dr. J.J.	53049	HG02.616
Schouten, Dr. T.E.	53175	HG02.512
Souren, Dr. A.F.M.M.	52269	HG02.814
Tretmans, Dr. ir. G.J.	52069	HG02.513
Vaandrager, Prof. dr. F.W.	52216	HG02.524
Vissers, Dr. G.A.N.	52686	HG02.830
Vliet, Prof. dr. M. van	53049	HG02.616
Vos, Drs. ing. P.M.	13026	TvA1 02.26
Weide, Prof. dr. ir. T.P. van der	53361	HG06.621
Welters, Drs. R.P.M.M.	11354	
Wiedijk, Dr. F.	52649	HG02.542

### 4.3 Procedure for “Schakelverklaringen”

This information is intended for students who enter the master programme on the basis of a Bachelor's degree from a Polytechnic (“HBO-doorstromers”). Such students need to complete a set of courses from the bachelor programme covering their deficiencies (“schakelprogramma”) before being able to register as master students.

For reasons of planning, the courses of the deficiency programme are intertwined with the courses of the master programme, which means that you are in fact allowed to take part in courses of the master programme before actually completing the deficiency programme. However, *you are not entitled to start work on your master's thesis until you have completed your “schakelprogramma” and are officially registered as master student.* In order to register as master student, you need to obtain a so-called “Schakelverklaring” from the Education bureau. Here's how (in view of the fact that HBO-doorstromers are usually Dutch, we shall describe this in Dutch below).

#### **Procedure voor schakelverklaringen:**

Je levert bij het onderwijsbureau de volgende gegevens in:

- naam-, ru-email-, adres- en opleidingsgegevens.
  - Het bewijs dat je ingeschreven staat voor de bacheloropleiding Informatica. Daarvoor lever je een kopie van je collegekaart in.
  - Een uitdraai van je cijferlijst. Deze vraag je op bij de facultaire studentenadministratie (HG0.134) en is voorzien van handtekening en stempel van de FSA (openingstijden ma-do: 13-16 uur, vrij 9-12 uur). Geef zelf even aan om welke cursussen het gaat (aanvinken of markeren met stift).
  - Je kunt je gegevens inleveren bij het onderwijsbureau. Als er niemand aanwezig is dan kun je je gegevens in de houten inleverkast doen. Het wijst zich vanzelf welke dat is.
2. Het onderwijsbureau controleert je gegevens, overlegt met de examencommissie en bereidt de verklaring voor.
  3. Je krijgt de verklaring binnen twee weken thuis gestuurd.
  4. Met deze verklaring moet je zelf bij de centrale studentenadministratie (Comeniuslaan 4) je inschrijving omzetten. **JE KUNT PAS MET JE Afstuderen BEGINNEN ALS INSCHRIJVING IS OMGEZET EN JE ALS MASTERSTUDENT STAAT GEREGISTREERD.**

## 4.4 Index of courses

Advanced Programming.....	25
Analysis of Embedded Systems.....	18
Business rules.....	20
Cognition and Representation.....	21
Complexity theory.....	22
Computational Intelligence.....	23
Design of Embedded Systems.....	24
Foundations of Information Systems (IS2).....	26
ICT & Samenleving 2.....	27
ICTproject in a different culture.....	29
Image Processing.....	19
Information Retrieval.....	28
Proof Assistants.....	30
R&D: Research 1.....	36
R&D: Research 2.....	37
R&D: Research 3.....	38
R&D: System Development.....	31
R&D: System Development Management 1.....	32
R&D: System Development Management 2.....	33
R&D: System Development Research.....	35
Reliability of Software Systems.....	39
Security in organisations.....	40
Security seminar.....	41
Semantics.....	42
Software Security.....	43
Testing techniques.....	44
Type theory.....	45
Wiskunde (schakelcursus).....	46