Faculty of Science

Prospectus 2009 - 2010

Computing Science

Master

Radboud University Nijmegen

Preface

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

This prospectus has been made with care. Nevertheless it is possible that it contains som inaccurasives and the authors can not 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
July 2009

Contents

1 General Information

1.1 Introduction

Welcome to the Institute for Computing and Information Sciences. The institute is part of the Faculty of Science of the Radboud University Nijmegen, and is responsible for the academic programmes in Computing Science and Information Sciences. At our institute you can follow a Bachelor programme (3 years) and/or a Master programme (1 year for Information Science; 2 years for Computing Science). 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 two-year Master of Science programme in Computing Science constitutes the follow-up to the Bachelor programme, and provides students with a thorough knowledge in computing science. 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'.

The master programme is taught in English (in the case of Dutch students only, courses may be taught in Dutch).

New

Students, who apply for their master exam **are not allowed to get an unsatisfactory mark** on their list of marks. An unsatisfactory mark is a mark lower than 6.0. This rule applies for students enrolling in the Master programme for the first time as of 31 August.

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 a post Polytechnic-degree (Dutch: HBO-diploma) can be qualified, although they are always obliged to do an extra, transition or conversion 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" (no unsatisfactory mark allowed from this year on). 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. For information on procedures, programmes you may send an email: hbo-info@cs.ru.nl.

Enrollment

The Central Student Administration takes care of all student enrollments at Radboud University Nijmegen. If you are already a student at the Radboud University, re-enrollment is done via Internet. In the months prior to re-enrollment, you will receive further information on this. In order to obtain your student and registration cards on time, you are requested to arrange your re-enrollment immediately after 1 June. You will receive your student card and registration card on average six weeks after your enrollment has been processed. Further information on enrollment can be found in the Student Statute and on the site www.ru.nl/studenten.

If you are graduating, you arrange your de-registration with the University Certification Bureau. For any questions about enrollment, please go to the Central Student Desk.

In the remainder of this chapter you will find some practical information and an overview of important dates, names and addresses. More information can be found on the website of the institute, www.ru.nl/iii/.

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 Office

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 Office. The staff members of the education office are listed further on in this chapter (see "Important names and addresses"). The coordinator of studies for Information Sciences is Vera Kamphuis, V.Kamphuis@cs.ru.nl and the coordinator of studies for Computing Science is Yella Kleijnen, Y.Kleijnen@cs.ru.nl.

The secretary's office of the Education Office can be found in the Huygens building, room HG02.540. 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 www.ru.nl/iii/ where subpages with practical information can be found.

Timetable and course information

When and where a course or an examination is planned can be found through our time table viewer on www.ru.nl/iii/onderwijs/rooster/ (English explanation can be found on www.ru.nl/iii/onderwijs/english_pages/time_table_generator). With this timetable viewer you are able to compose your personal weekly timetable for your individual combination of courses. Indicate the courses that you want to attend and you will be able to view your personal timetable for the actual week. From this academic year on the Radboud University provides time table information though a new programme. Further information follows in August/September.

Communication

A lot of communication in our institute is done electronically. Lecturers use Blackboard, e-mail or wiki in their courses. Also, the Education Office uses Blackboard and e-mail for important announcements regarding timetables 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-email address. *It is your responsibility to make sure that this address always remains intact*

But, if you're not enrolled in time (meaning, not before 15 September) it is possible that your name is not on the list of so called "active" students. We are not able to enroll "not active"students, and therefore you have to enroll yourself for the Blackboard community *NIII-onderwijs*.

Important names and addresses

Important names and addresses of various people and committees in our institute can be found in the appendix (chapter 4.2). Also, you will find there the contact details of the **master advisor,** Dr. Theo Schouten (T.Schouten@cs.ru.nl), the advisor for HBO-students, Dr. Hanno Wupper (h.wupper@cs.ru.nl) and the **master thesis coordinator,** Dr. Patrick van Bommel (pvb@cs.ru.nl).

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 first 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.*

For the next opportunity you have to ***register yourself*** *via KISS/TIS.*

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 certificate elsewhere: bachelor certificate (or kandidaats certificate)
- only for students who obtained their bachelor 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.

Make sure your grades have been processed. You ***must check this yourself***. ***New*** **for students enrolling the master programme on 31 August 2009 for the first time: unsatisfactory marks are *not allowed* on your mark list.**
If you register for your examination and not all the results have been processed, you will fail your examination and have to re-apply.

The regulations governing the examinations in August are somewhat different. For these, students can register up to May 31, 2010 and may do so even if several marks have not yet been obtained. These marks have to be delivered before August 31, 2010.

There are 11 examinations dates scheduled each year (usually the last Friday of the month, provided this is not an official holiday; in July there is no examination date at all). 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 Board.

Registration for the master exam is only possible for those students who have obtained their bachelor certificate and who are registered as a master student. It is not possible to register for both your bachelor exam and your master exam at the same time.

Rules on Teaching and Examination of the Master programme

The examination regulations have been laid down in two documents. The Education and Examination Regulations (OER) govern the general organization and scope of education and examinations. More specific regulations can be found in the Rules and Guidelines of the Examination Committee.

The full text of the OER can be found on the Internet (www.ru.nl/iii/examencommissie). The official document is in Dutch, but for the convenience of foreign students a translation is provided (same site).

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 “Studentenstatuut”, on www.ru.nl/studenten.

The examination board looks into matters only if you provide it with the following information:

- your name
- student number
- study programme
- which curriculum
- if it concerns a specific course
  - name of the course
  - exam code
  - name of the lecturer
  - Spring of Fall semester
- details of the request

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 'KISS Services' 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 office of the Institute for Computing and Information Sciences to communicate important information (on matters regarding education) 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 The Statutes for Students at the Radboud University

The student statutes consist of a description of the rights and responsibilities of all students registered at the Radboud Universiteit, based on statutory and university regulations. The student statutes and its appendices are on the Internet: www.ru.nl/studenten.

1.6 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 scholarschip, 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, Prof.Th. van der Weide (Th.P.vanderweide@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/).

Study trip: ICT in a different culture

Our institute take special interest in contributing to the development of ICT in other cultures. Our staff members have been involved in lecture programmes in countries like Ghana, South-Africa and Uganda, and students have the opportunity to take part in study trips abroad. Countries that have been visited thus far include South Africa, Uganda and India (see studiereis.cs.ru.nl/ and www.ict4kids.nl/ for some reports and additional information in Dutch).
Such extracurricular activities are embedded in courses which are not part of the compulsory programme, but which can be taken as part of the specialisation or free choice in either the Bachelor programme (Community Outreach Project) or the Master programme (ICT in a different culture). A course description of the latter course can be found in chapter 3 of this prospectus.

1.7 Finding a job

BBB

Job prospects for students of Computing or Information Science are excellent; many students already find a job before they graduate. Companies are keen to employ students with an academic career in IT-related disciplines. Every year the 'BBB' ("Bèta Bedrijven Beurs" organizes a job-market where companies present themselves to students. This annual career-event helps undergraduate and graduate students scout the job-market.

The BBB-event takes place in Spring semester at the Science Faculty (for the next event see www.bbb.science.ru.nl/): A great number of companies, organizations as well as follow-up degree programmes present themselves. Companies are present with a display and give lectures. You can gather information and talk with recruiters. Senior and PhD students can apply on-line around the time of the exhibition and stand a chance to be invited by one or more of the companies for an interview. These interviews are organized by BBB a few weeks after the exhibition. The chances to be invited at that moment are much higher as compared to when you send an open application to a company.

The exhibition is renowned for its casual atmosphere and for its service to visitors. Admission is free, no registering is needed and everybody receives the BBB-career guide.
Prior to the exhibition, BBB organizes workshops on a variety of topics that are relevant for job-seekers and career-starters, such as: interview training, case studies, but also more light-hearted topics.

Contact address: Heyendaalseweg 135, HG00.154, 024-3652388, www.bbb.science.ru.nl/, e-mail: bbb@science.ru.nl

To provide additional assistance in applying for jobs, the central Students Affairs Office at Comeniuslaan 4-6 also offers courses in presentation and has various facilities for job-orientation. More information can be found at their website, www.ru.nl/studenten/na_je_studie/informatiecentrum/ (website in Dutch).

1.8 Calendar 2009-2010

Academic year

August 31 2009 - August 31 2010

Semester

Fall semester: 31 August 2009 - Friday 29 January 2010;
Spring semester: 1 February - 16 July 2010.

Holidays

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 31-08-09	-	-	-
Fall Break*	26-10-09 / 30-10-09			
Christmas holidays	21-12-09 / 01-01-2010	-	-	-
Carnival	15-02-2010 / 19-02-2010	-	+	-
Eastern	02-04-2010/ 05-04-2010	-	-	-
May holiday	30-04-2010 / 07-05-2010	-	+	-
Queen's Birthday	30-04-2010	-	-	-
Liberation Day	05-05-2010	-	-	-
Foundation Day	20-05-2010	-	-	-
Ascension day	13-05-2010	-	-	-
day after Ascension day	14-05-2010	-	-	-
Whitsun	24-05-2010	-	-	-
Summer holidays	19-07-2010 / 27-08-2010 19-07-2010 / 13-08-2010 16-08-2010 / 20-08-2010 23-08-2010 / 28-08-2010	- - -	- + +	- - +

Quarters:

Quarter 1:	30-08-09 till 06-11-09
Quarter 2:	09-11-09 till 29-01-10
Quarter 3:	01-02-10 till 16-04-10
Quarter 4:	19-04-10 till 16-07-10

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

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 (in Dutch: propedeuse) concerns foundation courses. The courses are generally taught in Dutch. 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 programme takes 2 years. You will specialize in one of the research related themes of the Institute of Computing and Information Science. The courses are taught in English. Upon completing your studies, you will receive your master's degree and you may call yourself *Master of Science (MSc)*.

The Master programme contains several components: Basic courses, Variant related subsidiary subjects, Specialization (24 ec), Research and Development (20 ec), Free space (amount of ec is 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 you choose one:

- *Communication variant* (C): trains students in the direction of science communication, internally or towards society.
- *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.
- *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 a academic perspective.
- *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.

Within the Research variant it is possible to compose a master programme Computer Security. This is a two-year international programme. Students have to travel on Monday and Friday because the courses will be taught at three universities, Eindhoven, Nijmegen and Twente (TU/e; RU; UT).
More information on this programme can be found in chapter 2.3.

Also within the Research variant an interested student may choose the master programme Mathematical Foundations of Computer Science. More information can be found in chapter 2.3.

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.

compulsary courses			
Year 1 Fall semester	***ec***	***Year 1 Spring semester***	***ec***
R&D: Research 1	8	R&D: Research 2	6
		ICT & Society 2	3
Year 2 Fall semester	***ec***	***Year 2 Spring semester***	***ec***
R&D: Research 3 *or* R&D: System Development Research	6	Final Thesis	30

Optional Courses year 1-2	
Specialisation/theme	24
subsidiary subject	25
extra courses in Computing science	12
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.

compulsary courses			
Year 1 Fall semester	***ec***	***Year 1 Spring semester***	***ec***
R&D: System Development	8	R&D: System Development Management 1	6
		ICT & Society 2	3
Year 2 Fall semester	***ec***	***Year 2 Spring semester***	***ec***
R&D: System Development management 2	6	Final Thesis	30

Optional Courses year 1-2	
Specialisation/theme	24
extra courses in Computing science	12
subsidiary courses in Management (see below)	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 prepares students for a job in communication. It is not meant to be a preparation as a research PhD student in Computing Science. More information can be obtained at: www.filosofie.science.ru.nl

compulsary courses			
Year 1 Fall semester	***ec***	***Year 1 Spring semester***	***ec***
R&D: System Development (version 1) OR R&D: Research 1 (version 2)	8	R&D: System Development Management 1 (version 1)	6
		ICT & Society 2	3
Year 2 Fall semester	***ec***	***Year 2 Spring semester***	***ec***
R&D: System Development Research (version 2)	6	Final Thesis	30

Optional Courses year 1-2	
Specialisation/theme	24
Extra courses in Computing science	12
Subsidiary Subject Communication	21
Extra courses in Communication	6
free choice	10

Subsidiary subject in Communication (21 ec):

Year 1

- Introduction Science Communication (3 ec)
- Science & Social Interaction (3 ec)
- Risk Communication (3 ec)
- Boundary work (3ec)

Year 2

- Framing Knowledge (3 ec)
- Knowledge Society (3 ec)
- Science, Media and Strategy (3 ec)

2.3 Specialisation in Computing Science

Introduction

Computer systems nowadays penetrate all facets of society and personal life (ambient computing). Sometimes these systems appear in an easily recognizable form, as in personal computing, on-line shops such as Amazon, and pocket calculators, but also increasingly hidden, as in television sets, multimedia devices and so on. Computing may be pervasive in this modern digital age, but the associated software has revealed problems regarding security (breach of information access restrictions or privacy), reliability (the system does not behave as required), safety (use of the system is harmful), trustworthiness (low reliability of system services), efficiency (the system is unable to handle problems of a particular size) and alignment with work in practice. The inherent ever growing complexity of computer artifacts together with the usual slow pace of software development with its high costs and competitive pressures complicate matters further. These problems are due to a combination of inadequate modeling and engineering practices and of reluctance (and lack of obligations) to invest in solutions. Scientifically justified and cost-effective techniques for abstract modeling and analysis are a first requirement.

The mission of ICIS is to improve software development fundamentals by developing formal, mathematically founded theories, methods and tools to support the specification, design, analysis and evaluation of computer-based systems.

Research aims include improving the quality of software, with an emphasis on reliability, security, architecture and system alignment. Research is concentrated within several research groups, each with its own research focus and aims. These research groups function both as organisational units, and as areas of research.

On top of the group structure the institute works on several research themes, which typically involve more than one group.

Research at ICIS is concentrated in three themes: Model Based System Development, Digital Security and Intelligent Systems.

Research and Specialisations (Themes)

Specialisation in your master programme is related to one of the research groups and the courses are grouped around these specialisations, as you can see in the overview below.

Theme Foundation
** Proof Assistants (6 ec)*
** Complexity Theory (6 ec)*
** Semantics and Domain Theory (6 ec)*
** Type theory and Proof Assistants(6 ec)*

Theme Embedded Systems
** Testing techniques (6 ec)*
** Analysis of Embedded Systems (6 ec)*

** Design of Embedded Systems (6 ec)*
** Introduction to computer graphics (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 Systems (6 ec)*
** Testing techniques (6 ec)*
** Analysis of Embedded Systems (6 ec)*

Theme Artificial Intelligence
** Bayesian and decision Models in AI (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 programme 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 programme, 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 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. This so called conversion programme has priority over the master programme.

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 or Peter van Rossum (petervr@cs.ru.nl). See also the website at www.kerckhoffs-institute.org/.

Mathematical Foundations of Computer Science (MFoCS)

Throughout the centuries there has been a fruitful and mutually inspiring interaction between physics and mathematics. A similarly fruitful and exciting interaction has existed right from the start between computer science and mathematics. This ranges from the use of mathematics to model the foundations and explore the potentials and limits of computer science to the use of computers to help solve mathematical problems with a discrete component. This Research Master Program places itself squarely on this exciting and quickly developing interdisciplinary edge of deep theoretical developments.

In this Research Master Program, mathematicians working in areas pertinent to (theoretical) computer science, like algebra and logic, and theoretical computer scientists, working in areas as formal methods and theorem proving, join forces to establish a master program in the Mathematical Foundations of Computer Science, (MfoCS).

The emphasis of this program is on a combination of a genuine theoretical and up-to-date foundation in the pertinent mathematical subjects combined with an equally genuine and up-to-date training in key aspects of theoretical computer science. For this reason, the mathematics courses in this curriculum concentrate on Algebra, General Topology, Logic, Number Theory, and Combinatorics. The computer science courses concentrate on Formal Methods, Type Theory and Theorem Proving.For this master program we solicit students with a bachelor in mathematics or computer science that have a strong mathematical background and theoretical interests. We will select students based on their motivation and their background.

Programme

It is intended that students of this master program obtain a broad knowledge and understanding over a wide range of material in mathematics and theoretical computer science, bringing them in contact with the research frontier of the field. Consequently, the curriculum consists of both lectures (with exercise classes) and of research projects, which are organized in a Research Seminar and a Research Lab. There are 6 fixed courses for all students and the rest can be chosen from a list of elective courses.

Semester 1		**Semester 2**	
3 x 6 ec fixed	18	3 x 6 ec fixed	18
6 ec Elective	6	6 ec Elective	6
6 ec Kaleidoscope*	6	6 ec Research Seminar **	6
Total	30	*Total*	30
Semester 3		**Semester 4**	
3 ec Philosophy	3	Master Thesis	25
2 x 6 ec Elective	12	6 ec Elective	6
Master Thesis	15		
total	30	*Total*	31

* Kaleidoscope consists of selected topics of semantics of programming languages, basic complexity theory and rings and fields. Ths course will provide a crash course of some of the basic knowledge that we assume the students to be familiar with when starting this programme.

** In Research Seminar, the various teachers of this programme will give a short introduction to their research and present some research projects that the students can select one from for a small project.

For detailed information on fixed and elective courses, consult the website: www.cs.ru.nl/~herman/onderwijs/theorymaster and/or contact prof. Herman Geuvers: H.Geuvers@cs.ru.nl.

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. What you like. It is also possible to do your subsidiary subject at another university, or even abroad.

The subsidiary subject needs approval of the examination committee (examencommissie@cs.ru.nl).

Examples of (once chosen) subsidiary subjects (partly in Dutch):

Artificial Intelligence
Cognition and Representation (6)
Introduction to language and speech technology (6 ec)
Practicum Kennisverwerking voor expertsystemen (6 ec)
Introduction to pattern recognition(6 ec)

Security in Society
Information Law (7 e) or Knowledge and entrepeneurship (3 ec)
Dynamics of law (7 ec)
Criminology (7 ec)
Introduction to Language and Speech Technology (10 ects)

Social Psychology
Attitudes (6 ec)
Sociale Cognitie (6ec)
Sociale Psychologie (3 ec)
Reclame Psychologie (6 ec)
Sociale beïnvloeding in de praktijk (6 ec)

Management
Organisation Theory (5 ec)
Business and Society (5 ec)
Financial Management (5 ec)
Research Strategy Management (3 ec)
Science and Entrepeneurship (3 ec)
Information and Control (5 ec)

2.5 Final thesis

Master thesis

In your *master's thesis*, you will show that you are able to analyse a problem in Computing Science at master level and design a solution for this problem using scientific methods and techniques. It is possible to combine the research for the master's project with an internship in a suitable company.

On the site of the Master Thesis Lab (www.ru.nl/iii/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 Science. 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) if you need more information.

3 Course descriptions

Analysis of Embedded Systems

Course ID: **I00154**	*6 ec*	second semester	prof. dr. F.W. Vaandrager dr. J. Schmaltz

Study investment

- 30 hrs lecture
- 4 hrs personal study counseling
- 96 hrs student project
- 38 hrs individual study period

Introduction

Motivation/Overview: 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 traditional approaches of testing and simulation. It becomes essential to build mathematical models of these systems, and to use (algorithmic) verification methods to analyse these models. During recent years there has been enormous progress in the areas of hardware and software verification. 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 models and how to analyze them using state-of-the-art techniques and tools.

Objectives

After successful completion of the course, participants are able to:

- recognize situations in which the applications of model checking techniques for specification and analysis may be useful,
- explain the modelling frameworks and basic theory of finite state, real-time, and probabilistic automata,
- model (critical parts of) embedded systems as networks of automata,
- formalize desired properties of these systems in terms of automata or temporal logic, and
- use state-of-the-art tools for analysis of realistic embedded system problems.

Subjects

Theory: (symbolic) model checking, temporal logic, timed automata: difference bounded matrices, binary decision diagrams, bounded model checking, probabilistic automata, discrete-time Markov chains, Markov decision processes.
Tools: Uppaal, SAL, PRISM
Applications: Various controller synthesis and resource allocation problems from the embedded systems area, real-time operating systems, distributed algorithms and protocols: leader election for networks with ring, tree or general topology, mutual exclusion algorithms, communication protocols for physical and datalink layer.

Teaching methods

Participants are supposed to spend approximately 168 hours (=6ec) on this course: during 16 weeks they will need about 4 hours per week to attend+prepare classes (2 hours lectures, 1 hours problem session, 1 hour reading or working on exercises at home). In addition they will need about 35 hours for each of the three larger homework assignments (to be made in small groups of 2 or 3 participants) based on which their participation in the course will be evaluated. Participants are expected to be present during the lectures and the problem sessions.

Examination

Instead of a final exam this course has "integrated examination". Grades will be awarded on the basis of three larger homework assignments, which participants may do on their own of in groups of at most three. The final score is obtained as the average of the scores for the three assignments. If the final score is 5 or below, participants will be offered the opportunity to make a 4th homework assignment. The final score will then be the average of the 3 highest scores (so the lowest score may be dropped).

Prerequisites

Familiarity with propositional and predicate logic, automata theory, 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 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.

Literature

The course material consists of hand-outs, sheets, and recent papers from the literature. These will be made available electronically via the course page or distributed during the course.

Website

www.cs.ru.nl/~fvaan/PV/

Extra information

This course is part of the Embedded systems theme.

Bio-inspired Algorithms

Course ID: **IMC014** *3 ec* second semester dr. E. Marchiori

Study investment

- 32 hrs lecture

Introduction

Biologically inspired algorithms is a category of algorithms that imitate the way nature performs.
This category has been quite popular, especially for solving complex optimization problems. In this course you will learn and implement modern bio-inspired algorithms and use them to solve complex optimization problems.

Objectives

At the end of the course the student will:

- use state-of-the art biologically inspired algorithms, such as ant colony optimization and memetic algorithms.
- develop programming and methodology skills by solving real-life complex problems using bio-inspired algorithms.
- enhance communication and team capabilities by working in groups on a given project.
- further develop scientific writing skills by writing a paper.

Subjects

Topics considered in the course include:

- methodology for tackling complex optimization problems,
- ant colony optimization,
- memetic algorithms,
- multi-objective optimization.

Teaching methods

The course consists of two parts: theory and practice.
In the theory part bio-inspired methods are studied, while in the practical part the students will develop and implement a method based on bio-inspired algorithms for solving a given complex optimization problem.

Examination

Assignments and project design/implementation

Literature

Material will be made available during the course.

Website

www.cs.ru.nl/~elenam/bio_algorithms_vak.html

Bioinformatics

Course ID: **IMC013** *3 ec* second semester dr. E. Marchiori
dr. T.M.H. Dijkstra

Study investment

- 32 hrs lecture

Introduction

In bioinformatics the data are typically multidimensional and contain noise as well as technical and biological variability. Furthermore the underlining biological mechanisms to be analyzed are in general unknown and have a complex nature. Therefore models must be learned from the data.

This makes bioinformatics an interesting and challenging application area for computer science. In this course, the student will study issues and selected topics in bioinformatics and will learn to develop and apply computational approaches and algorithms for tackling these problems.

Objectives

- At the end of the course the students will be able to:
- model bioinformatics problems by means of advanced machine learning techniques;
- use proper methodologies for applying machine learning techniques to bioinformatics;
- avoid common pitfalls in biological data analysis;
- experience with computational and efficiency issues related to the implementation of algorithms
- to solve problems in bioinformatics;
- design and program effective methods based on feature selection, classification, prediction and data mining for bioinformatics;
- enhance current state-of-the-art tools for biological data analysis by means of machine learning techniques.

Subjects

Bioinformatics topics of the course include

- genome analysis,
- metagenomics,
- comparative analysis of biological networks,
- comparative protein sequence analysis.

Computational methods analyzed and applied to bioinformatics problems include

- optimization techniques,
- feature selection,
- unsupervised and supervised learning.

Teaching methods

The course consists of two parts. The first part will provide the bioinformatics background needed to tackle one of the projects proposed in the second part of the course. The project consists in the development and implementation of a method for tackling a specific bioinformatics problem.

Examination

Assignments and project design/implementation.

Literature

Survey papers and material made available during the course.

Website

www.cs.ru.nl/~elenam/bioinformatics_vak.html

Business Rules

Course ID: **IMK001** *6 ec* second semester dr. S.J.B.A. Hoppenbrouwers
prof. dr. ir. M.J. Plasmeijer

Study investment

- 32 hrs lecture
- 32 hrs problem session
- 104 hrs individual study period

Introduction

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

- Taxation laws.
- Rules governing the application of mortgages.
- 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 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.

A related field withing enterprise engineering is Business Process Modelling (BPM). BPM and the Business Rule Approach are interrelated and are arguably moving in each other's direction. This is why the BR course includes a specialised component concerning BPM. The approach we follow is comparative between BRA and BPM.

Objectives

After attending this course, students are able to:

- Position and value BRs and BPM as approaches within Information Systems and Enterprise Engineering
- Position and value various basic techniques and standards concerning BRs and BPM
- Perform basic specifications in some specific languages related to BPM (iTasks) and Rule-based Systems (CLIPS)
- Conceive, develop and refine original and well-founded ideas and argumentations concerning various aspects of BRs/BPM and their application in enterprises

Subjects

- Business Rules
- BPM
- SBVR and other standards
- Formal rule specification
- Tools and Engines
- Applications of BPM/BRs

Teaching methods

- About 12 intensive and interactive lectures, including guest lectures
- A small number of "discussion lectures", for which you have to prepare in advance
- About 6 practical sessions in which you can practice and execute practical specification assignments

Examination

50% A short research paper, addressing a relevant topic of choice (from a shortlist)
50% Practical exercises concerning the formal specification of rules

Prerequisites

- Some introduction to logic (for example Beweren & Bewijzen)
- Domain Modelling or similar knowledge

Website

Blackboard

Category Theory

Course ID: **WM033C** *6 ec* first semester prof. dr. B.P.F. Jacobs

Introduction

Category theory provides an abstract language for mathematics that concentrates on the essentials, namely "objects" and "morphisms" between them. The language of categories is elegant and remarkably powerful. It arose in the 1940s algebraic topology but is now used in many branches, not only of mathematics but also of theoretical computer science and physics. With a "categorical mindset" one recognises the structural aspects in a partical context, which helps to guide further investigation.

The course will introduce the basic concepts, for mathematicians, computer scientists and phycisists.

Objectives

The student will learn to use the language of categories, the basic constructions, and how to recognise and use them in various contexts.

Teaching methods

Lectures and tutorials

Examination

Written exam, at the end of the course.

Prerequisites

A certain level of mathematical maturity is required. Courses like topology, rings and fields, logic, universal algebra, lattice theory are useful.

Literature

Steve Awodey, Category Theory, Oxford Univ. Press, 2006.

Cognition and Representation

Course ID: **I00054** *6 ec* first semester dr. J.J. Sarbo

Study investment

- 30 hrs groepsgewijs college
- 30 hrs lecture
- 4 hrs personal study counseling
- 104 hrs individual study period

Introduction

The term "representation" usually refers to formalization, including a deductive or inductive use of formalized knowledge. This view, maintained by computer science, is opposed to the interpretation of knowledge by cognitive theory, according to which it is an expression of thoughts by a human observer. In this course we learn how these two conceptions of knowledge, formal and meaningful, can be linked with one another through a cognitively based model of knowledge representation. In addition, we learn how the proposed representation can be used for a uniform modelling of knowledge in different domains.

Objectives

- Making acquaintance with a semiotic concept of signs.
- The definition of a model of cognitive activity.
- Introduction of a model for logic, language, and mathematics.
- Learning the differences between formal and human interpretation.
- Using the cognitive model for problem elicitation.

Subjects

- Signs and interpretation
- Conceptualization processes
- Processing schema
- Problem elicitation as a conceptualization process
- Naive logical interpretation
- Syntactic language model
- Aplication in reasoning and mathematics
- Application in Text summarization

Teaching methods

This course makes use of problem directed education. The students are individually working on weekly exercises. A full solution of the exercises are developed in class.

Examination

A midsemester and a final test, both must be 5.5 or higher. The final grade is defined by the average of the two tests. This test grade can be adjusted by the average grade for the weekly exercises, but only if the latter is above the test grade (otherwise, no correction is applied).

Prerequisites

Basic knowledge in propositional logic, as well as in deductive and inductive reasoning (including mathematical induction) is required.

Literature

The Lecture Notes of the course are distributed via Blackboard.

Website

osiris.cs.kun.nl/~janos/CR.html

Extra information

This course makes use of problem directed education.

Compiler Construction

Course ID: **IMC004** *6 ec* second semester prof. dr. ir. M.J. Plasmeijer
dr. J.E.W. Smetsers

Study investment

- 56 hrs computer course
- 28 hrs lecture
- 14 hrs problem session
- 70 hrs individual study period

Introduction

No course description is available yet for this course.
Please contact the lecturer, Prof. M.J. Plasmeijer for more information:
R.Plasmeijer@cs.ru.nl.

Complexity Theory

Course ID: **I00023** *6 ec* second semester drs. P.J.B. van Rossum

Study investment

- 32 hrs lecture
- 32 hrs problem session
- 104 hrs individual study period

Introduction

This course is about the complexity of algorithms, that is, the costs: fast programs; division into classes; upper and lower bounds; computation models. A large part of this course is classical stuff that every computer scientist is supposed to know. However, some more specialist matters will be treated; various important techniques and insights will be presented by studying a number of nice and interesting subjects. Let us mention:

- Computer arithmetic with applications, e.g, in cryptography
- Gaining speed at the expense of memory space;
- What everybody should know about polynomials (applicationsa.o. in cryptography) ;
- nice and clever constructions of logical circuits;
- how for can we compress a string?
- ... etc...

Complexity theory is useful, e.g, for courses in the realm of computeralgebra (math. dept.) and telecommunication; and of course if you wish to do a Complexity Master.
Implicitly and explicitly, complexity theory contains fundamental knowledge and techniques that apply to the whole of Informatics. It is nice. It is beautiful.

Objectives

After this handsome course the student has:

- mastered a number of important notions, tricks and algoritms from complexity theory, building upon what (s)he has learned in the bachelor complexity course.

More in particular this means winning

- knowledge of, and
- insights in, and
- operational dexterity in problem solving regarding

subjects such as or connected with those as summed up below.

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.
- Arithmetic on the computer, with some applications in cryptography
- Depending on time, some small subjects.

Teaching methods

Depends on the number of participants. Usually, each week there are two lectures of one hour (if there are few students these may be of an informal character); with weekly exercises. Besides there are two reserved assistance hours on individual basis; personally or via the internet.
Examination is in principle orally, or in the form of some small project. (If the number of students is "large", in mutual agreement there might be a written exam.)

Examination

In principle oral exam, on the basis of handed in exercises.

Prerequisites

Our basic bachelor courses on math, programming and theory (formal languages, logic, complexity).

Literature

There are course notes that can be downloaded. To be found together with more information about CXT at: www.cs.kun.nl/~bolke/Teaching.html

Website

www.cs.ru.nl/~bolke/CXT/BeschrCXT.html

Computer Algebra

Course ID: **WM069B** *6 ec* second semester dr. W. Bosma

Introduction

The aim of the lectures will be to provide an introduction into the area of computer algebra. The main focus will be on algebra and algorithms, but there will also be some attention to complexity and implementation issues.

On the one hand this should give some insight into the underlying mathematics, on the other hand also some ability to use computer algebra systems will be acquired. This should lead to an understanding of the theoretical possibilities and the practical limitations of computer algebra.

Among others, topics will be algorithms for efficient integer, rational and modular arithmetic, and computing with polynomials, rational functions and power series, determining the factorization and common factors of integers and of polynomials over finite fields or the integer ring, as well as some techniques from linear algebra and algebraic geometry.

Examination

By mutual agreement

Prerequisites

Linear algebra, groups, rings and fields

Bayesian and decision models in AI

Course ID: **IMC012** *6 ec* second semester dr. P.J.F. Lucas
dr. A.J. Hommersom
dr. N. Carvalho Ferreira

Study investment

- 24 hrs lecture
- 4 hrs personal study counseling
- 6 hrs laboratory course
- 20 hrs student presentation
- 26 hrs student project
- 10 hrs problem session
- 78 hrs individual study period

Introduction

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. The new name of the course is: Bayesian and decision models in AI

Objectives

At the end of this course, the student should be able to:

- know the cognitive aspects of reasoning with uncertainty
- understand the mathematical 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

Subjects

- Introduction to Computational Intelligence
- Cognitive aspects of uncertainty reasoning
- Early models of uncertainty
- Probability theory
- Bayesian networks: principles
- Markov independence
- Reasoning with Bayesian networks

- Building Bayesian networks
- Learning Bayesian networks
- Decision making

Teaching methods

lectures, seminar, tutorials, practical assignment

Examination

Written exam in addition to seminar presentations and practical work.

Prerequisites

Course "Intelligent Systems"

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 and T. Nielsen, Bayesian Networks and Decision Graphs, Springer, New York, 2007.

Website

www.cs.kun.nl/~peterl/teaching/CI/

Extra information

The course is part of the AI Masters and also suitable for AI students.

Design of Embedded Systems

Course ID: **I00155** *6 ec* first semester dr. J.J.M. Hooman

Study investment

- 32 hrs lecture
- 4 hrs personal study counseling
- 40 hrs student project
- 16 hrs problem session
- 76 hrs individual study period

Introduction

A large part of the functionality of many devices is realized by software. The development of this embedded software is usually far from trivial. For instance, because it has to interact with its hardware environment, sensor information is not perfect, it has to satisfy real-time requirements, there are memory or power limitations, or faults have to be tolerated. This course addresses the development of embedded software, from specification to implementation. The emphasis is on hands-on experience by means of exercises on a real-time operating system and a concrete case study with real-time and fault-tolerant aspects.

Objectives

Students are able:

- To reason about schedulability of real-time tasks and to devise appropriate scheduling strategies.
- To develop small applications on a real-time operating system, using the available primitives for scheduling, mutual exclusion and communication.
- To apply a systematic development method to design a relatively small embedded system using appropriate tools.

Subjects

- Characteristics of embedded systems
- Scheduling theory
- Real-time operating systems
- Requirements definition
- Design decisions
- Deployment of software on hardware
- Use of UML for embedded software
- Development tools

Teaching methods

There is a strong emphasis on active participation by the students and getting hands-on experience with the main concepts such as scheduling, real-time operating systems and the development of embedded software. Central activity of the last part of the course is the realization of a case study in a systematic, iterative, way. Important are regular discussions on the design and frequent feedback on the iterations. The used techniques and tools are introduced briefly in the course, but mainly internalized by the application to the case study.

Examination

The final exam of this course consists of:

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

In addition, the results of the exercises on scheduling and real-time operating system will be taken into account. The final score is the mean of the score for the scheduling and OS exercises and the score for the case study. Both parts of the course must have been completed sufficiently.

Prerequisites

Basic programming skills using the programming language C.

Literature

Relevant literature will be distributed during the course.

Website

www.cs.ru.nl/~hooman/DES/

Extra information

This course fits in the theme Embedded Systems.

Advanced Programming

Course ID: **I00032** *6 ec* first semester prof. dr. ir. M.J. Plasmeijer
dr. P.W.M. Koopman
dr. P.M. Achten

Study investment

- 32 hrs lecture
- 64 hrs laboratory course
- 16 hrs problem session
- 56 hrs individual study period

Introduction

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

Objectives

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

Subjects

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

Examination

Written, closed book.

Prerequisites

Bachelor Computer Science.

Literature

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

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

Course ID: **I00035** *6 ec* second semester dr. P. van Bommel

Study investment

- 32 hrs lecture
- 32 hrs problem session
- 104 hrs individual study period

Introduction

In this course we study foundations of information systems in detail. These fundamental aspects will be organized around the notion of *transformation of information models*. We will specify the *syntax and semantics* of concrete transformations. This will lead us to a framework for *reasoning* about transformations, in which different design strategies can be considered, for example quality-driven design strategies. Our study of foundations has a theoretical nature, but practical cases will also be considered.

Objectives

You will learn:

- transformation of information models
- reasoning about transformations
- optimization of transformations

Subjects

- Basic *information language* with a suitable representation mechanism.
- *Complexity* of the representation mechanism.
- *Wellformedness conditions* for representations, with a parameterized generation algorithm.
- *Correctness proof* of generation algorithms.
- Rule-based *population transformation* and operation transformation.
- Distinction between generation operators and *mutation operators*.
- Application of these operators in *(interactive or automated) design processes*.
- *Predicting and comparing the behaviour* of different transformation strategies.

Examples of transformations are the following. *Data format* may change when it is transferred between systems, including changes in data structure, data model, data schema, data types. *Interpretation of data* may vary when it is passed on from one person to another. Changes in interpretation belong to data semantics rather than data structure. *Level of detail* may change when exchanging data between departments or organizations, for example going from co-workers to managers or from local authorities to central government. *Systems development phase* of data models may vary, for example when implementation-independent models are mapped to implementation-oriented models.

Examination

Individual student paper (p) and central written examination (e). Every student must write a paper. This paper will be 20% of the final result. The final result (f) is computed as follows: IF e<5 OR p<5 THEN f = min(e,p) ELSE f = 0.2*p + 0.8*e.

Prerequisites

It would be good if you have some experience with information models, for example the relational model or the entity-relationship model (or a similar model). Furthermore, it would be good if you can handle basic mathematical definitions.

Literature

- The *lecture notes*.
- Instructions for the *student paper*.
- *Handbook* of data modelling (from 1959 until 2008).
- Suggestions for *further reading* (not required).

Lattice Theory

Course ID: **WB050C** *6 ec* first semester prof. dr. M Gehrke

Study investment

- 28 hrs lecture
- 28 hrs problem session

Introduction

We will introduce lattices both as partially ordered sets and as algebras. From the partial order point of view, we treat Hasse diagrammes, complete lattices, Galois connections and Formal Concept Analysis. From the algebraic point of view, we treat the homomorphism theorems, special classes such as modular, distributive, and Boolean algebras as well as the representation theory for finite lattices and its relation to classical propositional logic.

Objectives

After this course the student is familiar with the notions of ordered algebraic structures, including lattices and Boolean algebras, and has seen the connection to various topics in algebra, analysis, and computer and information science. He or she is familiar with Formal Concept Analysis and Birkhoff Duality for finite distributive lattices, and has seen several applications including one to knowledge representation.

Teaching methods

- 28 hours tutorial
- 28 hours lecture

Examination

Written exam.

Prerequisites

Ringen en lichamen 1

Literature

B.A. Davey & H.A. Priestley, *Introduction to Lattices and Order*, 2nd edn (CUP 2002).

Hardware and Operating Systems Security

Course ID: **IMC001** *6 ec* first semester dr. ir. E. Poll
W.I. Mostowski

Study investment

- 4 hrs excursion
- 4 hrs groepsgewijs college
- 30 hrs lecture
- 8 hrs personal study counseling
- 2 hrs student presentation
- 80 hrs student project
- 42 hrs individual study period

Introduction

Hardware plays an important role in securing computer systems. This is most obviously the case for smartcards, but mainstream operating systems also rely on hardware features, as do mobile phones, set-top boxes, ATMs, etc. In the future secure hardware looks set to play an bigger role, eg. in controversial) Trusted Computing initiatives.

The focus of the course is on smartcards, as an example of a complete platform, consisting of hardware and an OS, which has been specifically designed to provide security, and for which both the techniques for attacks and defences are highly evolved.

Objectives

At the end of the course students can

- explain how smartcards, RFIDs, and hypervisors work, and explain their security objectives, and the hardware, principles, and techniques used to achieve these;
- explain currently known attacks on smartcards and associated countermeasures;
- build a simple smartcard application from scratch, incl. the associated protocol design and key management.

Subjects

Topics:

- Smartcards: hardware, operating systems, attacks (logical, power analysis (SPA, DPA), fault injection, invasive attacks) and countermeasures;
- RFID tags: idem;
- Sample applications, such as banking/EMV, e-passports, ov-chipkaart, GSM;
- Trusted Platform Modules (TPM) and Trusted Computing (TC);
- Hypervisors and microkernels.

Teaching methods

Weekly lectures and project work.

As part of the project work, we will design and build a small smartcard applications; this includes thinking about security requirements, coming up with a design and being able to motivate this, coming up with the necessary protocols and key management, implementing this, and getting it to run on actual JavaCard hardware.

Examination
The course will be evaluated on the basis of the project work.

Prerequisites
You are expected to have good knowledge of cryptography and have some Java programming skills.

Website
www.cs.ru.nl/~erikpoll/hw

Extra information
This course is part of the Kerckhoffs security master specialisation.

ICT in a different culture

Course ID: **IMI001** *6 ec* first semester prof. dr. ir. T.P. van der Weide

Study investment

- 32 hrs lecture
- 64 hrs student project
- 72 hrs individual study period

Introduction

This course is an international broadening course to explore levels, practice and opportunities and threats for Information and Communication Technology in a non-western context.

Each year, some specific context is chosen, referred to as the country of interest. This context is further refined by specific research projects.
For 2009/2010 we will visit India.

Objectives

As a result of active participation in the student study tour and a substantive study of literature, the student will be able for the specific situation in the country of interest:

- To describe the overall economic situation, and the governmental policy relating to it. Furthermore also the situation on micro economic level; trained and untrained people, employees of small businesses, unemployed and people living in rural and urban areas.
- To describe the educational level of the schools and universities, and the governmental policy towards it. Also the policy towards people without access or with difficult access to education, and its corresponding educational level.
- Outline the different levels of demand of the market, and also how these levels compare to the Dutch educational levels.
- To indicate the opportunities for Computing Science and Information Science research for both universities and industry.
- To indicate current ICT utilisation in universities, government and industry both in the urban as well as in the rural areas. Hereby focusing on the type of demand, and to what extent the current level of access can answer this demand.
- To describe the current status of the infrastructure in the country of interest with special emphasis on availability of hardware (distribution and communication channels, supply of parts), opportunities for maintenance and available knowledge and skills (at several levels including the Dutch vocational levels MBO and HBO).
- To be able to motivate to your own insight which types of development work have the desired effect, and how they correlate with governmental policies.

After successful completion of the course, students are capable to provide a motivated answer on questions regarding the role of ICT in the country of interest, focusing on the one hand on usefulness and sensibility issues and on the other hand on feasibility restrictions and opportunities in the context of

- industry (in what sectors and what problems can be expected?)
- government (idem)

- university (idem)
- society (idem)

This should also includes cultural and other societal aspects of relevance.

Subjects

- Understanding cultural differences
- International cooperation
- In-depth study of country to be visited
- Research methods
- Competency development
- ICT in different cultures

Teaching methods

The course is organized around a number of small research projects that preferably are supplied by companies participating in this course. Companies are invited to participate in this course, and to contribute interesting research projects.

The course consists of three phases:

- Initial phase. The students formulate their research question, and prepare the research to be done during the visit in the country of interest. Also they prepare themselves son the specific cultural background of that (part of the) country. During this phase, the organization of the concluding seminar is set up.
- The actual research phase: 10 days research in destination country.
- Evaluation phase. During this phase, the students will finish their research report. Furthermore, they will organize a seminar in which the results are presented. Some extra activities are also organized during this seminar, such as guest speakers and a forum discussion.

The students work in groups on their research projects. At regular moments they provide feedback on the performance of the other members in their group. This feedback will be used during the assignment of an individual mark for each participant. After the initial phase there will be an intermediate mark. A positive intermediate mark is required to participate in phase 2.

The students will also be part of organization tasks such as the organization of the seminar, p.r. activities, creating a professional combined research evaluation report for internal and external distribution.

This contribution will also be part of the mark of each student.

Examination

The students write a short paper about their research. Furthermore, the students are marked for their contribution to organizational activities in the context of this course (such as organizing the final seminar).

Literature

The students will receive material during the course.

Website

studiereis.cs.ru.nl/

Informatics and Society 2

Course ID: **I00037** *3 ec* second semester dr. L. Consoli

Study investment

- 32 hrs lecture
- 6 hrs personal study counseling

Introduction

The course Informatics and Society 2 (Informatica en Samenleving 2) explores cultural and social issues that have been made possible by the development of information technology Starting from the concepts of privacy and 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. We will analyze among others the cultural meaning of hackerism, the role different ethical frameworks play in discussing technological advances, and the social/ethical/cultural implications of artificial intelligence (AI).

Objectives

The student will:

- be acquainted with a number of philosophical and ethical theories, in general and applied to his/her object of study;
- be able to recognize the implicit presuppositions in a number of scientific advances of his/her object of study;
- be able to reflect on the normativity of his/her object of study;
- be able to articulate his/her reflection in a number of short papers and a research paper.

Subjects

- ICT & the human nature
- Ethical theories
- Ethics of ICT (hackerism as case study)
- The mind-body problem
- Philosophical foundations of Artificial Intelligence

Examination

Students will have to write 4 short essays and a final term paper. There is no final examination planned.

Literature

Literature references and reading material will be made available through Blackboard.

Website

https://blackboard.ru.nl

Information Retrieval

Course ID: **I00041** *6 ec* second semester prof. dr. ir. T.P. van der Weide
prof. dr. ir. W. Kraaij

Study investment

- 30 hrs lecture
- 34 hrs problem session
- 104 hrs individual study period

Introduction

Finding relevant documents no longer seems to be the major challenge of state-of-the-art search engines. Where recall and precision were major concerns in the early days of their existence, trying to convey information rather than just data seems to be a major concern nowadays. Offering a long list of documents in order of their relevance score is known to be a too simple interface.

In order to improve on this, solid knowledge of the information retrieval problem and its main techniques is imperative. As there are still many questions about the essentials, a strong relation with ongoing research activities is indispensable.

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

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

An important application area is the World Wide Web.

Objectives

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

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

Subjects

The course consists of three main parts:

- Fundamentals
  1. After a discussion on the problem areas of Information Retrieval,
  2. the evaluation methods for Information Retrieval are discussed.
  3. The Boolean model is discussed, together with techniques related with inverted list document representation.

   4. The vector model is the most used model. As a method for knowledge extraction, the singular value decomposition (main component analysis) is discussed.
   5. The probabilistic model applies Bayesian learning techniques to Information Retrieval.
- Knowledge extraction and Information processing
   1. Query languages in relation with cognitive aspects of information searching.
   2. Autonomous query improvement techniques (global context analysis).Guided query improvement techniques (feedback).
   3. Pseudo relevance feedback (local context analysis).
   4. Clustering techniques for knowledge extraction
- Exploiting link structure and access data on the Web
   1. Web retrieval.
   2. Exploring the reference structure between documents (for example, PageRank).
   3. Exploring document appreciations (collaborative techniques).
   4. Clickdata analysis, learning to rank
   5. Special topics contributed by the participants

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

Teaching methods

- The course is divided in parts, each part is concluded with a test.
- Each week there are 4 contact hours, in which the new material is presented and exercised. Several problems sessions involve exercises with IR software tools.
- The participants have to make a contribution to the course (see below).

Student contribution

Participants have to choose a topic from the most recent SIGIR conference or TREC conference.These contributions will be centered around special themes in Information Retrieval. The themes will vary from year to year. The actual themes will be announced during the lectures.

The students make an extended summary of the topic chosen, and present this during the lecture.The contributions are peer reviewed by the participants of the course.

Examination

Written exam in addition to presentation and practical work.

Prerequisites

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

Literature

Lecture notes will be made available via Blackboard.

Website

blackboard.ru.nl/

Introduction to Computer Graphics

Course ID: **NM058B** *6 ec* Fall semester drs. P.F. Klok

Study investment

- 40 hrs computer course

Introduction

During the lectures the basics of computer graphics and image processing are covered. The assignments of the practical course follow the topics of the lecture text. A written examination about the topics of the lecture text concludes the course.

Objectives

Familiarize with basic terms and techniques of computer graphics and image processing and implement various techniques in both fields. Through the knowledge and the practical experience of this course, well-funded judgement on graphics matters and quick mastering of graphics software should be obtained.

Subjects

- **General:**
  basic notions, synthetic camera, windows, viewports, clipping, coordinate systems, graphical standards
- **Interactive Graphics:**
  windows, graphical objects, input classes, user interface
- **2D and 3D Graphics:**
  transformations, projections, graphics pipeline, hidden-line and hidden-surface removal
- **Raster Graphics:**
  frame buffers, scan conversion, colour models
- **Rendering:**
  ray tracing and ray casting, reflections, shading, splines
- **Image Reconstruction**:
  fourier transforms, backprojections
- **Image Enhancement:**
  filtering, histogramming
- **Visualization:**
  pseudo colour, lookup tables

Teaching methods

- 28 hours lecture
- 132 hours practical work

Examination

Exercises and preliminary examination.

Prerequisites

Required is programming experience at the level of Programmeren (NB021B) or equivalent

Literature

Necessary:

- Website www.hef.ru.nl/~pfk/education/icg/

Strongly recommended:

- Foley, van Dam, Feiner, Hughes, Phillips, *Introduction to Computer Graphics*, Addison Wesley, 1993, ISBN 0-201-60921-5.

Website

www.hef.ru.nl/~pfk/education/icg/

Extra information

Lectures followed by practical work to elaborate on lecture topics
The graphical package OpenGL is used with programs written in the C programming language

Intro Pattern Recognition

Course ID: **NB054B** *6 ec* fourth quarter prof. dr. H.J. Kappen
dr. W.A.J.J. Wiegerinck
prof. dr. T.M. Heskes

Study investment

- 13 hrs lecture
- 13 hrs problem session

Introduction

Statistical pattern recognition provides methods to recognize complex relations in noisy data. These problems arise in many areas such as visual object recognition, analysis of genetic data, financial data or neuroscience data. The course provides an introduction to pattern recognition from a Bayesian point of view and treats various classical methods (such as linear discriminant classification) as well as more recent methods (such as mixture models, the EM method, neural networks, backpropagation and Bayesian learning methods).
This course in intended for master students in computer science, AI or cognitive science. For more information about the course see also www.snn.ru.nl/~wimw/collegePRforAI.html. Physics or Math students that are interested in this topic are advised to follow the course Neural Networks and Information Theory in the Bachelor phase and the course Machine Learning in the Masters' phase.

Objectives

- The student is capable to develop a probabilistic model for regression, classification and density estimation. The student has access to a wide range of model classes, such as Gaussian models, Gaussian mixtures, generalized linear models and feedforward neural networks
- The student is capable to formulate how the model can be trained using the principle of maximum likelihood estimation as well as using the full Bayesian approach
- The student is capable to derive efficient algorithms to optimize and train these models and is capable to implement these methods in Matlab

Subjects

See www.snn.ru.nl/~wimw/collegePRforAI.html

Teaching methods

- 28 hours lecture
- 28 hours tutorial

Examination

- Written exam
- Additional bonus for problem sessions (“werkcolleges”) (1 point max.)
- Additional bonus for short computer practicum assignments (1 point max.)

Prerequisites

- For students mathematics or physics: Linear Algebra, Calculus, Kansrekening
- For students AI or computer science : Wiskunde voor AI or equivalent

Literature

- Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press
- exercises through the course website.

Website

www.snn.ru.nl/~wimw/collegePR.html

Law in Cyberspace

Course ID: **IMC006** *6 ec* first semester dr. ir. E. Poll

Study investment

- 12 hrs lecture
- 24 hrs problem session
- 132 hrs individual study period

Introduction

Short description:
Can I legally play a dvd bought in the US on a European dvd player? Is downloading music from the internet stealing? Is using an open WiFi access point legally permissible? What about liability when illegal activities are conducted by someone else than the owner of the access points? Do I have a right to be anonymous on the Internet?
These are but some of the legal questions arising in the network society. In this course, provided by researchers of TILT - the Tilburg
Institute of Law, Technology, and Society - an introduction is provided to the role of law with regard to ICT. Focusing on problems
arising in everyday life online, the course will provide an overview of these issues from a legal perspective.

Subjects

The course will start with a brief introduction to law, regulation and the defining features of core two legal areas: private law and criminal law. This is followed by lectures on important areas within Cyberlaw:

- privacy, data protection, and identity management
- copyright and 'code' as code
- computer crime I
- computer crime II
- electronic signatures and e-commerce.

Examination

written exam

Extra information

Precise information will be available in September 2009.

Proof Assistants

Course ID: **I00139** *6 ec* second semester dr. F. Wiedijk

Study investment

- 30 hrs lecture
- 30 hrs laboratory course
- 108 hrs individual study period

Introduction

This course is an introduction to proof assistants. A proof assistant is a program in which one can encode a mathematical proof (which often is a proof in which software or hardware is proved correct) in such a way that it will be impossible for it to be incorrect, as the computer will check full correctness.

The course both presents all main proof assistants, and also teaches two proof assistants in detail: the Mizar system from Poland, and the HOL system from Cambridge in the UK.

The course focuses on the use of proof assistants for their use in mathematics.

Proof assistants are one of the main subjects studied in the Intelligent Systems research group of the ICIS institute.

Objectives

After the course students can:

- develop a formalization in a proof assistant
- implement a simple proof assistant
- select an appropriate proof assistant for a given purpose
- install and use a proof assistant that they have not seen before

Subjects

- the Mizar proof assistant
- the HOL proof assistant
- the LCF-architecture for proof assistants
- defining and proving in a proof assistant
- dependent types and implicit arguments
- clusters and registrations, redefinitions
- searching for lemmas in a proof assistant
- proof automation and decision procedures
- goals, tactics, tacticals
- conversions and rewriting
- applications of proof assistants

Teaching methods

The course has three parts:

- Mizar: using a proof assistant
- HOL: implementing a proof assistant
- overview of the other proof assistants

The first two parts consists of lectures. During these parts students will be working on exercises during the practicum. There will be a final exercise for each of these two parts, that will have to be handed in.

The third part consists of presentations by the students. Each student will present a different proof assistant. As preparation for this part there is one lecture with an overview of proof assistant, to enable students to make a motivated choice for their presentation.

If there are only a few students, the practicum will not be given as contact hours, but instead there will be regular meetings for feedback.

Examination

The final note will be the average of three marks:

- the mark for the Mizar formalization
- the mark for the small LCF-style proof assistant
- the mark for the presentation in the third part of the course

There is a deadline for handing in the two exercises. If this deadline is not met, the exercise will be considered not to have been made.

Prerequisites

Predicate logic. Basic set theory. Lambda calculus.

Literature

There are various documents that will be used to support the course. These will be made available on the website in pdf format

The main of these documents are course notes on the Mizar system by Freek Wiedijk, and the manual of the HOL Light system by John Harrison.

Website

www.cs.ru.nl/~freek/courses/pa-2010/

Extra information

This course is very interesting for mathematics students. This course does not teach the Coq proof assistant, as that is the subject of the Type Theory and Proof assistants course.

R&D: System Development

Course ID: **I00156** *8 ec* first semester prof. dr. M.C.J.D. van Eekelen
drs. G.F.M. Paulussen

Study investment

- 28 hrs lecture
- 4 hrs personal study counseling
- 136 hrs student project
- 56 hrs individual study period

Introduction

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

Objectives

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

Teaching methods

This course has an intensive practical project. A so-called GipHouse project. Within GipHouse projects are performed in a simulated software enterprise.
see also https://lab.cs.ru.nl/gip/ and www.giphouse.nl.

Examination

project evaluations, projectreport, exam on the literature. If both project and theoretical exam are satisfactory (6.0 or higher) the final result is the weighted average of the two, otherwise the final result is the lowest one of the two.

Prerequisites

SE, SO1

Literature

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

Website

https://lab.cs.ru.nl/gip/ and www.giphouse.nl

Extra information

project, report

R&D: System Development Management 1

Course ID: **I00157** *6 ec* second semester dr. T.E. Schouten

Study investment

- 16 hrs lecture
- 112 hrs student project
- 40 hrs individual study period

Introduction

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 (2EC) and a (4EC) 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 software house in which the students of this course perform roles as: Project manager, Quality manager, Contract Owner, Public Relations Manager, Director. These roles can be adjusted depending on the specific situation of a given semester (e.g. number of students). All students work, within the management structure of GiP-House, under the supervision of the director, with the aim to create an effective and efficient software house management structure. The managers use, if necessary, (internal or external) experts.

Objectives

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

Subjects

For the theoretical topics, see the website:

- project management
- metrics
- scheduling and tracking
- capability maturity model
- component based development
- implementation
- team management, projectorganization
- requirement analysis

Teaching methods

There will be about 8 presentations of 2 hour each.
Further working as a manager for about 112 hours as a manager in GiPHouse.

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 examination for the theoretical part.
- The examination for the theoretical part might be in two ways:
  - A 2 hour written examination, this determines the final grade. No literature may be consulted during the examination.
  - A paper, this also determines the final grade.
- The decision between the two ways is decided at the beginning of the course and is based on the previous education of each student

Prerequisites

Bachelor Computing or Information Science

Literature

Software Engineering A practitioners Approach: European Adaptation, sixth edition , Roger S. Pressman
Sheets of the presentations

Website

www.cs.ru.nl/~ths

Extra information

Many students entering with a HBO bachelor will already have experience as manager in software projects. If that experience is sufficient to have obtained the goals of the practical part of this course, the teacher can exempt the student from the practical part. The procedure to obtain this will be indicated by the teacher at the beginning of this course.

R&D: System Development Management 2

Course ID: **I00158** *6 ec* first semester prof. dr. M. van Vliet

Study investment

- 32 hrs lecture
- 46 hrs student project
- 90 hrs individual study period

Introduction

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.

The course consists of a practical part (GiPHouse) and a theoretical part. In the theoretical part the students have regular classes and they work on an integral case study which is woven into the course.

Objectives

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. 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:

- To work as a manager in a team with clearly defined tasks, deliverables and constraints.
- To work independently with regard to software development and to assess and improve the quality of the process.
- To guide, assess and implement the most suited software development methodologies for the given case.
- To be able to guide the future development of a software-house.
- To select and implement the most appropriate software development tools for the given case.
- To select and implement the best system-architecture (programming languages, software development tools, hardware, middleware, and communication components) for the given case.
- 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).
- To be able to recognize communicative and organizational issues in a software-house environment and to effectively act upon that.
- 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.

- 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.
- To work at an academic level: to be able to assess one's own performance and the ability to improve upon that.
- To be able to work independently on one's own professionalism: to gain and create knowledge based upon own literature survey and gained advice.

Subjects

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 resembles the phase in an IT career in which the manager takes responsibility for the management of a software-house. 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 one's own actions and the corresponding achieved results is a very important aspect.

Teaching methods

Within GiP-house the student aims at two goals: realising their own personal development goals, and delivering a high quality system. In this structure the focus is very much on team-work in order to realise the required deliverables.

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.

Prerequisites

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

Literature

- Course books form the GiP-House library.
- Course material handed out during the course.
- Online GiP-House handboek.(www.cs.kun.nl/is/edu/gip/hb/inhoudsopgave.html).
- GiP-House website (www.giphouse.nl).

Extra information

For further information with regard to the GiP-House, please contact Dr. Theo Schouten at ths@cs.ru.nl.
SDM2 can be combined with courses from the Faculty of Management (www.ru.nl/nsm/).
Within the practical component of SDM2 the student can perform the following roles: **Projectmanager, Qualitymanager, Contract Owner, Public Relations Manager, Director**. These roles can be adjusted depending on the specific situation of a given semester (e.g. number of students. All students work, within the management structure of GiP-House, under the supervision of the director, with the aim to create an effective and efficient software house management structure. The managers use, if necessary, (internal or external) experts.

R&D: System Development Research

Course ID: **I00162** *6 ec* first semester prof. dr. M.C.J.D. van Eekelen

Study investment

- 28 hrs lecture
- 4 hrs personal study counseling
- 80 hrs student project
- 56 hrs individual study period

Introduction

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

Objectives

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

Examination

Project evaluations, projectreport, exam on the literature. If both project and theoretical exam are satisfactory (6.0 or higher) the final result is the weighted average of the two, otherwise the final result is the lowest one of the two.

Prerequisites

SE, SO1, Research Lab 1,2

Literature

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

Website

https://lab.cs.ru.nl/gip/ and www.giphouse.nl

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 reseach 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: **I00159** *8 ec* first semester dr. J.H. McKinna

Study investment

- 26 hrs lecture
- 4 hrs personal study counseling
- 194 hrs individual study period

Introduction

The course R&D: Research 1 (RDR1) is all about doing research.
The main question in this course is: when is research considered scientific? The answer is built around the ideas of formulating a good research question, embedding it in known research, using proper methods and using valid deduction techniques to draw conclusions.

Objectives

After the course the student is able to:

- Come up with a subject inside a given framework.
- Formulate a good research question.
- Write a research proposal that embeds the topic in a suitable context.
- Know how a research question is split in subquestions, tasks and deliverables.
- Know how to find scientific literature via the library.
- Perform a literature study.
- Combine the results found into a paper with a good structure.
- Write a review report about a scientific paper.

Subjects

The topic of the research itself is limited to some framework that might be different each year. Some of the topics that have been used in the past:

- XML
- Exotic algorithms

Within this framework the topic is basically free.

Some explicit topics of the course itself:

- How can you find scientific literature?
- What are good research questions?
- What are scientific methods for doing research?
- What is a good review report?

Teaching methods

During the course there are several general lectures where some theory is being explained. (Research methodology, library instruction, review process, etc.)

Furthermore, there will be some individual process meetings where students explain to the teacher the status of their research project.

But most of the time students work by themselves on their project. In groups of two. They have to write a proposal and actually perform the described tasks. Both the research proposal and the research results will be presented to the rest of the group.

Examination

The final grade is a weighted average of all assignments during the course with the following weight factors:

- Research proposal (15%)
- Presentation of proposal (10%)
- Draft paper (30%)
- Referee report (10%)
- Presentation of research results (10%)
- Final paper (25%)

Activity and/or absence during obligatory presentation session can have effect on rounding the grades,as can the late submission of work.

Prerequisites

No specific knowledge is required, but the normal rules for following a master's course apply.

Literature

Students have to find their own background literature for their specific topic.

Website

www.cs.ru.nl/~james/TEACHING/RDR1

R&D: Research 2

Course ID: **I00160** *6 ec* second semester dr. J.H. McKinna

Study investment

- 18 hrs lecture
- 6 hrs personal study counseling
- 144 hrs individual study period

Introduction

The course R&D: Research 2 (RDR2) builds upon the knowledge acquired within R&D: Research 1 (RDR1).
However, in RDR1 a literature study was good enough, but in RDR2 it is not. Students are expected to really bring their own contributions into the research.
The results need to be written down in a scientific paper and need to be presented.

Objectives

After the course the student should be able to:

- Come up with a good research proposal.
- Act on that proposal.
- Select appropriate scientific methods for answering the research question.
- Work together with an expert in the research area.
- Be able to put own results in the scientific context.
- Transform technical results into readable text.
- Present the research using slides.

Subjects

The topic for the research is free. Each student chooses his own supervisor within the department and together they come up with an appropriate topic. This idea is worked out into a research proposal.

Teaching methods

There is typically one lecture that is used to explain how this course works.
Students must present an initial written research proposal and slide presentation.
During the semester, they work with their supervisor and meet with the lecturer(s) for progress reviews.
Finally, they submit a final paper and give a final obligatory slide presentation.

Examination

Basically the supervisor advises the lecturer on a grade for your research project.
However, via an oral exam the student needs to prove that he did not only write the paper but is also able to talk about it. This can lead to a small adjustment of the grade recommended by the supervisor. Finally, activity and/or absence during presentation sessions can also adjust the grade slightly.

Prerequisites

You must have followed RDR1 seriously, which means that at least you should have handed in a draft paper.

Literature

Students have to find their own literature.

Website

www.cs.ru.nl/~james/TEACHING/RDR2

Extra information

Because we ask staff members to reserve time for supervising projects, we expect from students also that they reserve enough time for this course!

R&D: Research 3

Course ID: **I00161** *6 ec* first semester dr. J.H. McKinna

Study investment

- 10 hrs lecture
- 4 hrs personal study counseling
- 40 hrs student project
- 114 hrs individual study period

Introduction

The course R&D: Research 3 (RDR3) is divided into two separate tracks: the research track (R-track) and the supervisor track (S-track).
Within the R-track the student needs to follow their own research as in RDR2. This can provide the opportunity for example to undertake a preliminary feasibility study towards an afstudeer project, or to continue a line of research started in RDR2.
Within the S-track you will supervise a group of students that is working on an RDR1 project.

Objectives

If a student goes for the R-track, the goals and aims are basically the same as in RDR2. But the student has the opportunity to become more experienced in doing research.

After doing the S-track a student should be able to:

- Supervise a small research project.
- Identify pitfall in someone else's research project.
- Study papers well enough to be able to explain them to others.
- Write a semi scientific article about a given topic.
- Write a review report.
- Reflect on their achievements.

Subjects

In the R-track students choose their own topic, and should find an appropriate supervisor from the ICIS staff.
In the S-track the RDR1 students choose the topic, but the RDR3 students can have influence on it by helping with the research question.

Teaching methods

There are only a few lectures in this course. One for explaining how this course works and a few for theoretical background on e.g. the semi scientific article.
Since most students prefer the S-track there is usually only one presentation session of R-track projects.

Furthermore, both in R-track and S-track there are several individual progress meetings.

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.

Prerequisites

For doing the S-track it is needed that both RDR1 and RDR2 are completed.
For doing the R-track it suffices if you have completed either RDR1 or RDR2.

Literature

Some pointers to the writing of semi scientific articles are provided.
Students have to find their own literature concerning the research.

Website

www.cs.ru.nl/~james/TEACHING/RDR3

Extra information

If there are more RDR3 students opting for S-track than there are RDR1 projects, the teacher decides who can do S-track or who should do R-track.

Reliability of Software Systems

Course ID: **I00018** *6 ec* first semester prof. dr. M.C.J.D. van Eekelen

Study investment

- 32 hrs lecture
- 4 hrs personal study counseling
- 76 hrs student project
- 56 hrs individual study period

Introduction

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.

Objectives

- 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

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, Fortify, Esc-Java, Sparkle, PVS, Yasper, mcrl2, Gast, Torx.

Teaching methods

- 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.

Examination

Evaluation Report, End presentation

Prerequisites

Bachelor Computing Science

Website

https://lab.cs.ru.nl/laquso/

Security in organisations

Course ID: **I00153** *6 ec* first semester drs. P.J.B. van Rossum
prof. dr. E.R. Verheul

Study investment

- 32 hrs lecture
- 8 hrs personal study counseling
- 32 hrs laboratory course
- 96 hrs individual study period

Introduction

Information security deals with the preservation of the confidentiality, integrity and availability of information. The leading standard on information security is ISO 27001 that defines the notion of a Information Security Management System (ISMS). This is a means for the management of an organization to be in control of the information security risks. Fundamental within ISO 27001 is that information security is considered to be a 'process' and not a 'product' one can simply buy. The process allows management to ensure that others within their organization are implementing security controls that are effective.

One of the difficulties of the information security process is its multidisciplinary nature: it needs to grasp security requirements from the organization business processes (where the managers typically are not savvy on information security) and to translate them to security controls. These controls can be of various types, including ICT technical or cryptographic. Moreover, the process needs to check that the operational effectiveness of the chosen controls is satisfactory and to adapt the controls (or the surrounding framework leading to the controls) if required.

Within the course this process is explored both from a theoretical and a practical level never loosing sight of the computer science perspective. To this end the course also has several 'hands-on' exercises including conducting a Windows EDP audit, a network audit and a network penetration. The course provides the basic information on information security required by the security officer of an organization, by IT security auditors and by IT security consultants. As information security is still a rapidly evolving topic (some might argue it is even still in its infancy) the course can also provide inspiration for further scientific research.

Objectives

- Learn to control information security risks within an organization in an holistic fashion (procedural, organizational and technical).
- Getting familiar with the leading standards in this area, their shortcomings and practical implementation guidelines.
- To learn to map policies to technical countermeasures and vice versa.
- To learn how to write and enforce security policies.
- To learn some basic techniques in security auditing.
- Getting an idea of the practical aspects of information security.
- Getting inspiration for further scientific research.

Subjects

- International standards for information security and risk management
- Implementing information security and risk management
- Risk analysis methods
- Privacy
- Electronic signatures (law, practice, technical)
- EDP auditing
- Secure development and aquisition of software
- Business continuity management
- Network and database security
- Special topics: pseudonimization, phyiscal access control, digid, ideal

Teaching methods

The course consists of 2 hours of lectures per week and there is a lab session of 2 hours per week. Much of the course will be case-study based. Except to be doing a lot of background reading using the reader.

Examination

Mandatory assignments and mandatory written exam.

Prerequisites

The bachelor security course Security.

Literature

This course has a reader.

Website

www.cs.ru.nl/~petervr/secorg/

Extra information

Related courses:

- Software security
- Network security
- But also appropriate courses related to computers and law are an option.

Semantics and Domain Theory

Course ID: **IMC011**	*6 ec*	second semester	prof. dr. J.H. Geuvers dr. J.H. McKinna

For information contact: H.Geuvers@cs.ru.nl

Security Seminar

Course ID: **I00136** *6 ec* second semester dr. J.H. Hoepman
prof. dr. B.P.F. Jacobs

Study investment

- 20 hrs lecture
- 24 hrs student presentation
- 136 hrs individual study period

Introduction

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.

Objectives

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.

Subjects

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

Teaching methods

For this seminar, a handful of topics has been selected, and key scientific publications for each of these topics are provided. To participate, and to complete the seminar successfully, you have to do the following.

- Attend all lectures.
- Give a 2 hour lecture on a topic of your choice.
- Prepare a final paper on a *different* topic of your choice.
- Act as a referee for the final paper of another student.

The topics for the presentation and the student paper will be discussed and distributed among students during the first lecture.

Examination

Consists of 3 parts.

- giving a 2 hour presentation
- writing a research paper
- reviewing another research paper

The final grade is determined by a weighted average of the grades on these parts, provided that all grades are larger than or equal to a 5.5

Prerequisites

Students should have succesfully completed the Cryptography course.

Literature

Selected scientific publications; see course webpage.

Website

www.cs.ru.nl/~jhh/secsem.html

Extra information

For more information refer to the course website, www.cs.ru.nl/~jhh/secsem.html

Software Security

Course ID: **ISOFSE** *6 ec* second semester dr. ir. E. Poll

Study investment

- 4 hrs groepsgewijs college
- 30 hrs lecture
- 2 hrs personal study counseling
- 4 hrs laboratory course
- 2 hrs student presentation
- 40 hrs student project
- 86 hrs individual study period

Introduction

Bad software is probably the most important cause of computer security problems. This course is about the challenges in developing secure software and the technologies that can be used to improve software security, at the various stages in the software development life-cycle, and at various "levels", eg. specific to an individual application or at the level of the programming language.

Objectives

At the end of the course students

- can explain the common ways in which software security fails;
- are able to identify security objectives of applications and identify likely places where they might fail;
- can explain methods and technologies that can help in the development of secure software;
- can apply some of these techniques in practice.

Concrete examples of attacks and countermeasures are often specific to a certain setting (a programming language and/or type of application); the aim provide enough insight to be able to assess problems and proposed solutions in other situations.

Subjects

- Common security vulnerabilities, such as input validation problems (buffer overflows, SQL injections, etc.), race conditions, broken access control, XSS, CSRF, etc.
- Security measures in the software development life cycle: architecture, language/platform, implementation, testing, code review
- Language-based security: typing, (Java) sandboxing, untrusted code security
- Information flow
- (Tool-supported) Static Analysis
- Examples of advanced type systems, e.g. for alias control or information flow
- Program Verification and Proof-Carrying Code (PCC)

Teaching methods

Weekly lectures and project assignments.
The project work consists of assignments in which students analyse more or less realistic pieces of code for potential security flaws using various techniques and tools.

Examination

The final grade is based on a written exam and marks for the project assignments.

Prerequisites

Programming skills, in particular in C(++) and Java.

Literature

Selected articles on topics treated in the course are made available via the course webpage.
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.

which all available in the library.

Website

www.cs.ru.nl/~erikpoll/ss

Extra information

This course is part of the Kerckhoffs security master specialisation.

Testing Techniques

Course ID: **I00110** *6 ec* first semester dr. ir. G.J. Tretmans
dr. J. Schmaltz
G. Igna

Study investment

- 40 hrs lecture
- 3 hrs personal study counseling
- 55 hrs student project
- 10 hrs problem session
- 60 hrs individual study period

Introduction

Testing is part of almost any software development project. Systematic and effective testing is an important technique for quality assessment and control. Yet, testing is often an underexposed and underestimated part of software development. 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, 'state-of-practice' testing techniques as well as new developments, such as model-based testing, will be presented. Moreover, some guest lectures will give insight into the industrial practice of software testing.

Objectives

The goals of the course "testing techniques" are that students:

- obtain an overview of, and insight in the importance and the place of structured and systematic testing within the software development cycle;
- are familiar with standard testing concepts, terms, and nomenclature;
- know and recognize different kinds, phases, and aspects of structured testing;
- develop skills in applying some well-known techniques for developing tests;
- get experience in applying some test tools;
- have knowledge of, and can apply some of the latest research results in testing, in particular in model-based 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,
- testing equivalences,
- model-based testing with transition systems.

Teaching methods

The course consists of two parts. In the first part the state of practice and established testing techniques will be discussed (following the curriculum of ISTQB - the Int. Software Testing Qualifications Board). The second part will be about new developments in research, with emphasis on model-based testing. Both parts will consist of regular lectures, guest lectures, exercise lectures, and homework assignments.

Examination

Homework assignments (in groups of 2 students) and examination.
To pass the course:

- the average of homework assignments must be at least 5.0;
- the written examination must be at least 5.0;
- the final mark is the average of homework assignments and examination.

Prerequisites

Students are assumed to have some programming experience, and some experience with formal methods in software development. The course "testing techniques" is related to topics in programming and software engineering, courses in software quality, and to more formal topics such as "analysis of embedded systems".

Literature

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

Type Theory and Proof Assistants

Course ID: **IMC010** *6 ec* first semester dr. F. Wiedijk
prof. dr. J.H. Geuvers

Study investment

- 30 hrs lecture
- 30 hrs laboratory course
- 108 hrs individual study period

Introduction

This course is an introduction to type theory. Type theory studies the typed version of lambda calculus, and applies it to mathematics and computer science. It is one of the main foundations for the technology of proof assistants. In a proof assistant a mathematical proof (which often is a proof in which software or hardware is proved correct) can be developed with support from the computer in such a way that it has the highest reliability possible. The main proof assistant that is based on type theory is the Coq system from INRIA in France.

Objectives

After the course students can:

- explain the notions in the subjects list below
- make type derivations in the type systems lambda-arrow, lambda-P and lambda-2
- translate natural deduction proofs to lambda terms in these type systems, and vice versa
- relate detours to term redexes and eliminate detours from natural deduction proofs
- give the detours for the various logical connectives
- define simple inductive types and predicates and write functional programs in the Coq logic on them
- calculate the recursion principles for simple inductive types
- modify non-dependent and non-polymorphic datatypes into their dependent and polymorphic counterparts
- use lambda-P as a logical framework for a simple logic
- use the Coq proof assistant to make a non-trivial Coq formalization
- understand a basic research paper on type theory

Subjects

- the Coq proof assistant
- the type system lambda-arrow: simple type theory, simply typed lambda calculus
- the type system lambda-P: dependent types and logical frameworks
- the type system lambda-2: polymorphism and second order propositional logic
- Pure Type Systems in general and the Barendregt cube
- the Curry-Howard-de Bruijn isomorphism
- classical versus intuitionistic logic
- the type checking and inhabitation problems
- term normalization, detour elimination, and their relation
- inductive types and predicates, recursive functions
- program extraction
- a more advanced research topic

Teaching methods

The course has two parts.

The first part consists of working through the course notes by Femke van Raamsdonk. There will be lectures and there is a practicum in which exercises will be made (there are both Coq and pen-and-paper exercises).

In the second part a more advanced topic will be studied. There will be further lectures about a reseach topic, and after that a research article will be presented in parts by the students to the group.

Besides those two parts, each participant will create a Coq formalization. Each student will have a different subject for that formalization.

The practicum will use the ProofWeb system, which allows students to use the Coq proof assistant through the web. If there are only a few students, the practicum will not be given as contact hours, but instead there will be regular meetings for feedback.

Examination

The final note will be the average of three marks:

- the mark for the presentation in the second part of the course
- the mark for the final written exam
- the mark for the Coq formalization

There is a deadline for handing in the Coq exercise. If this deadline is not met, the Coq exercise will be considered not to have been made.

Prerequisites

Basic lambda calculus. Functional programming.

Literature

Course notes written by Femke van Raamsdonk of the Free University Amsterdam. These course notes will be available on the website in pdf format. One or more research articles about type theory, that also will be made available in pdf format.

Website

www.cs.ru.nl/~freek/courses/tt-2009/

Extra information

This course is interesting for mathematics students.

3.1 Courses MT-variant

MT-variant

For the course descriptions please consult the website studiegids.science.ru.nl/2009/en/science/prospectus/afstudeervariant_management_en_toepassing/courses/

Business and Society **FMT001C** *5 ec*	fall semester	dr. G.A.N. Vissers prof. dr. B. Dankbaar
Organization Theory **FMT002C** *5 ec*	spring semester	prof. dr. B. Dankbaar
Innovation Management **FMT003C** *5 ec*	fall semester	ir. L.J. Lekkerkerk
Strategy and Marketing **FMT004C** *5 ec*	fall semester	Dr. P.E.M. Ligthart dr.ir. N.G. Migchels
Finance and Accounting **FMT005C** *5 ec*	spring semester	drs. R.A. Minnaar

3.2 Courses C-variant

C-variant

For the course descriptions please consult the website:
studiegids.science.ru.nl/2009/en/science/prospectus/afstudeervariant_communicatie/courses/

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

4 Practical Information

4.1 Important names and addresses

Faculty of Sciences

Heyendaalseweg 135, 6525 AJ Nijmegen
Huygens building
tel.: 024-3616161 (Radboud University - general phone nr.)

Education Bureau for Computing and Information Sciences

General:
Secretary's office: HG02.540, Huygens building
tel.: 024-3652084

Staff:
Ms.mr. Resi Westerman, pr/secretary; R.Westerman@cs.ru.nl
Ms. Marcha Jelissen, pr/secretary; M.Jelissen@cs.ru.nl
Ms.drs.Vera Kamphuis, 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, pvb@cs.ru.nl
website: www.cs.ru.nl/mtl/

Education Board

Prof.dr.Herman Geuvers, director; H.Geuvers@cs.ru.nl
Dr. Sjaak Smetsers, coordinator master programme Computing Science; S.Smetsers@cs.ru.nl
*Prof.dr.Th.P.*van der Weide, coordinator master programme Information Science; Th.P.vanderweide@cs.ru.nl
Ms. Vera Kamphuis, MA, head of the education office; V.Kamphuis@cs.ru.nl
Ms. Yella Kleijnen, secretary, Y.Kleijnen@cs.ru.nl
Jos Groenewegen: assessor@cs.ru.nl
Xander Damen: assessor@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 Board

General e-mail account: examencommissie@niii.ru.nl
Website: www.cs.ru.nl/examencommissie/index.html

Coordinator of international affairs for Computing and Information sciences

Prof.dr. Th.P.van der Weide, Th.P.vanderweide@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/

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

info@thalia.nu (general info
website: www.thalia.nu/

Alumni association Ninja (for Computing and Information Sciences)

website: www.cs.ru.nl/ninja/

4.2 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 (transition programma or in Dutch "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 a **few** basic 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 (no unsatisfactory marks allowed for students entering this programme as of 31 august 2009):

- 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. Hierop mag geen onvoldoende staan. De cjferlijst 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 brievenbus naast de deur doen. Het wijst zich vanzelf welke dat is.
- Het onderwijsbureau controleert je gegevens, overlegt met de examencommissie en bereidt de verklaring voor.
- Je krijgt de verklaring binnen twee weken thuis gestuurd.
- 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.

5 List of lecturers

Name	Email	Phone (024-)	Room
Achten, Dr. P.M.	p.achten@cs.ru.nl	52483	HG02.616
Bommel, Dr. P. van	p.vanbommel@cs.ru.nl	52645	HG02.611
Bosma, Dr. W.	w.bosma@math.ru.nl	52311	HG03.716
Carvalho Ferreira, Dr. N.	nivea@cs.ru.nl	52104	HG02.620
Consoli, Dr. L.	l.consoli@science.ru.nl	53065	HG02.824
Dijkstra, Dr. T.M.H.	t.dijkstra@science.ru.nl	52633	HG02.544
Eekelen, Prof. dr. M.C.J.D. van	marko@niii.ru.nl	53410	HG02.074
Gehrke, Prof. dr. M	m.gehrke@math.ru.nl	53220	HG03.720
Geuvers, Prof. dr. J.H.	herman@cs.ru.nl	52603	HG02.526
Heskes, Prof. dr. T.M.	t.heskes@science.ru.nl	52696	HG02.524
Hoepman, Dr. J.H.	jhh@cs.ru.nl	52599	HG02.049
Hommersom, Dr. A.J.	arjenh@cs.ru.nl	52104	HG02.618
Hooman, Dr. J.J.M.	j.hooman@cs.ru.nl	52069	HG02.633
Hoppenbrouwers, Dr. S.J.B.A.	stijnh@cs.ru.nl	52645	HG02.611
Igna, G.	g.igna@cs.ru.nl	52123	HG02.634
Jacobs, Prof. dr. B.P.F.	b.jacobs@cs.ru.nl	52236	HG02.076
Kappen, Prof. dr. H.J.	b.kappen@donders.ru.nl	14241	0.12 M244
Klok, Drs. P.F.	p.klok@hef.ru.nl	52214	HG03.077
Koopman, Dr. P.W.M.	pieter@cs.ru.nl	52483	HG02.616
Kraaij, Prof. dr. ir. W.	w.kraaij@cs.ru.nl	52076	HG02.059
Lucas, Dr. P.J.F.	peterl@cs.ru.nl	52611	HG02.614
Marchiori, Dr. E.	elenam@cs.ru.nl	52647	HG02.521
McKinna, Dr. J.H.	james.mckinna@cs.ru.nl	52610	HG02.514
Mostowski, W.I.	w.mostowski@cs.ru.nl	52076	HG02.059
Paulussen, Drs. G.F.M.	g.paulussen@cs.ru.nl	52085	HG02.068
Plasmeijer, Prof. dr. ir. M.J.	rinus@cs.ru.nl	52644	HG02.621
Poll, Dr. ir. E.	e.poll@cs.ru.nl	52710	HG02.073
Rossum, Drs. P.J.B. van	petervr@letterboxes.org	52077	HG02.069
Sarbo, Dr. J.J.	janos@cs.ru.nl	53049	HG02.513
Schmaltz, Dr. J.	j.schmaltz@cs.ru.nl	52104	HG02.618
Schouten, Dr. T.E.	t.schouten@cs.ru.nl	53175	HG02.609
Smetsers, Dr. J.E.W.	s.smetsers@cs.ru.nl	52509	HG02.068
Tretmans, Dr. ir. G.J.	tretmans@cs.ru.nl	52069	HG02.633
Vaandrager, Prof. dr. F.W.	f.vaandrager@cs.ru.nl	52216	HG02.622
Verheul, Prof. dr. E.R.	eric.verheul@cs.ru.nl	52599	HG02.049
Vliet, Prof. dr. M. van	m.vanvliet@cs.ru.nl	53175	HG02.609
Weide, Prof. dr. ir. T.P. van der	th.p.vanderweide@cs.ru.nl	53361	HG06.621
Wiedijk, Dr. F.	freek@cs.ru.nl	52649	HG02.512
Wiegerinck, Dr. W.A.J.J.	w.wiegerinck@science.ru.nl	15040	0.24 M244

6 Index of Courses