This master’s programme emphasizes experimental computer science, rather than theory, and typically requires design, implementation, testing, and performance analysis of software for advanced parallel and distributed systems. It contains a balance between classes and practical work: about one third of the first three semesters consists of practical projects. The final semester is a master’s thesis, which will usually involve doing research in conjunction with one of the faculty members. In contrast to other master’s programmes, PDCS requires that students explicitly apply to be admitted. Up-to-date information can be found here.
<table>
<thead>
<tr>
<th>Inhoudsopgave</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>History, philosophy &amp; social aspects of science</td>
<td>1</td>
</tr>
<tr>
<td>Compulsory Optional Courses Theory Computer Science</td>
<td>1</td>
</tr>
<tr>
<td>Compulsory Optional Courses Programming</td>
<td>1</td>
</tr>
<tr>
<td>Optional Courses</td>
<td>1</td>
</tr>
<tr>
<td>Individuele vakken</td>
<td>2</td>
</tr>
<tr>
<td>Compulsory Courses</td>
<td>2</td>
</tr>
<tr>
<td>Vak: Advanced Logic</td>
<td>3</td>
</tr>
<tr>
<td>Vak: Advanced Selforganisation</td>
<td>4</td>
</tr>
<tr>
<td>Vak: Advanced Topics in Computer and Network Security</td>
<td>4</td>
</tr>
<tr>
<td>Vak: Advanced Topics in Distributed Systems</td>
<td>6</td>
</tr>
<tr>
<td>Vak: Cluster and Grid Computing</td>
<td>7</td>
</tr>
<tr>
<td>Vak: Coding and Cryptography</td>
<td>7</td>
</tr>
<tr>
<td>Vak: Computer and Network Security</td>
<td>8</td>
</tr>
<tr>
<td>Vak: Computer Graphics</td>
<td>9</td>
</tr>
<tr>
<td>Vak: Computer Networks Practical</td>
<td>10</td>
</tr>
<tr>
<td>Vak: Concurrency and Multithreading</td>
<td>11</td>
</tr>
<tr>
<td>Vak: Data Mining Techniques</td>
<td>12</td>
</tr>
<tr>
<td>Vak: Distributed Algorithms</td>
<td>13</td>
</tr>
<tr>
<td>Vak: Distributed Systems</td>
<td>14</td>
</tr>
<tr>
<td>Vak: Evolutionary Computing</td>
<td>15</td>
</tr>
<tr>
<td>Vak: Industrial Internship</td>
<td>16</td>
</tr>
<tr>
<td>Vak: Internet programming</td>
<td>17</td>
</tr>
<tr>
<td>Vak: Logical Verification</td>
<td>17</td>
</tr>
<tr>
<td>Vak: Master Project</td>
<td>18</td>
</tr>
<tr>
<td>Vak: Operating Systems</td>
<td>19</td>
</tr>
<tr>
<td>Vak: Operating Systems Practical</td>
<td>19</td>
</tr>
<tr>
<td>Vak: Parallel Programming</td>
<td>20</td>
</tr>
<tr>
<td>Vak: Parallel Programming Practical</td>
<td>21</td>
</tr>
<tr>
<td>Vak: PDCS Programming Project</td>
<td>21</td>
</tr>
<tr>
<td>Vak: Performance Analysis of Communication Networks</td>
<td>22</td>
</tr>
<tr>
<td>Vak: Research Proposal Writing</td>
<td>23</td>
</tr>
<tr>
<td>Vak: Scientific Writing in English</td>
<td>24</td>
</tr>
<tr>
<td>Vak: Selected Topics in Parallel and Distributed Computer Systems</td>
<td>25</td>
</tr>
<tr>
<td>Vak: Selected Topics in PDCS</td>
<td>26</td>
</tr>
<tr>
<td>Vak: Term Rewriting Systems</td>
<td>27</td>
</tr>
</tbody>
</table>
History, philosophy & social aspects of science

The choice of one of these elective courses, has to be discussed and agreed upon with the master coordinator or a personal mentor and approved by the Examination Board.

Compulsory Optional Courses Theory Computer Science

Compulsory choice Theoretical Computer Science of 6 credits, at least one choice out of the courses below.

Note: Every programme, including the choice of optional courses, has to be discussed and agreed upon with the master coordinator or a personal mentor and approved by the Examination Board.

Vakken:

<table>
<thead>
<tr>
<th>Naam</th>
<th>Periode</th>
<th>Credits</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding and Cryptography</td>
<td>Periode 1</td>
<td>6.0</td>
<td>X_405041</td>
</tr>
<tr>
<td>Concurrency and Multithreading</td>
<td>Periode 5</td>
<td>6.0</td>
<td>X_405064</td>
</tr>
<tr>
<td>Distributed Algorithms</td>
<td>Period 4</td>
<td>6.0</td>
<td>X_400211</td>
</tr>
<tr>
<td>Logical Verification</td>
<td>Ac. Year (September)</td>
<td>6.0</td>
<td>X_400115</td>
</tr>
</tbody>
</table>

Compulsory Optional Courses Programming

Compulsory choice Programming of 6 credits, at least one choice out of the courses below.

Note: Every programme, including the choice of optional courses, has to be discussed and agreed upon with the master coordinator or a personal mentor and approved by the Examination Board.

Vakken:

<table>
<thead>
<tr>
<th>Naam</th>
<th>Periode</th>
<th>Credits</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Networks Practical</td>
<td>Periode 5+6</td>
<td>6.0</td>
<td>X_405072</td>
</tr>
<tr>
<td>Internet programming</td>
<td>Periode 1</td>
<td>6.0</td>
<td>X_405082</td>
</tr>
<tr>
<td>Operating Systems Practical</td>
<td>Periode 5+6</td>
<td>6.0</td>
<td>X_405071</td>
</tr>
<tr>
<td>Parallel Programming Practical</td>
<td>Periode 2+3</td>
<td>6.0</td>
<td>X_400162</td>
</tr>
<tr>
<td>PDCS Programming Project</td>
<td>Ac. Year (September)</td>
<td>12.0</td>
<td>X_405054</td>
</tr>
</tbody>
</table>

Optional Courses

Opleidingsdelen:
- Individuele vakken

Vakken:

<table>
<thead>
<tr>
<th>Naam</th>
<th>Periode</th>
<th>Credits</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Logic</td>
<td>Periode 4</td>
<td>6.0</td>
<td>X_405048</td>
</tr>
<tr>
<td>Advanced Selforganisation</td>
<td>Period 2</td>
<td>6.0</td>
<td>X_400434</td>
</tr>
<tr>
<td>Computer Graphics</td>
<td>Period 2</td>
<td>6.0</td>
<td>X_400106</td>
</tr>
<tr>
<td>Concurrency and Multithreading</td>
<td>Periode 5</td>
<td>6.0</td>
<td>X_405064</td>
</tr>
<tr>
<td>Data Mining Techniques</td>
<td>Period 5</td>
<td>6.0</td>
<td>X_400108</td>
</tr>
<tr>
<td>Evolutionary Computing</td>
<td>Period 1</td>
<td>6.0</td>
<td>X_400111</td>
</tr>
<tr>
<td>Industrial Internship</td>
<td>Ac. Jaar (september)</td>
<td>6.0</td>
<td>X_405080</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Periode 4</td>
<td>6.0</td>
<td>X_405067</td>
</tr>
<tr>
<td>Performance Analysis of Communication Networks</td>
<td>Periode 1+2</td>
<td>6.0</td>
<td>X_400165</td>
</tr>
<tr>
<td>Scientific Writing in English</td>
<td>Ac. Jaar (september), Periode 4, Periode 5, Periode 6</td>
<td>3.0</td>
<td>X_400592</td>
</tr>
<tr>
<td>Selected Topics in Parallel and Distributed Computer Systems</td>
<td>Ac. Jaar (september)</td>
<td>3.0</td>
<td>X_400426</td>
</tr>
<tr>
<td>Selected Topics in PDCS</td>
<td>Ac. Jaar (september)</td>
<td>6.0</td>
<td>X_400379</td>
</tr>
<tr>
<td>Term Rewriting Systems</td>
<td></td>
<td>6.0</td>
<td>X_400121</td>
</tr>
</tbody>
</table>

Individuele vakken

Compulsory Courses

The courses
Advanced Topics in Computer and Network Security
and
Advanced Topics in Distributed Systems
and
Research Proposal Writing
and
Master Project PDCS
are part of the second year of the curriculum.

Compulsory alongside the below mentioned courses (78 credits), are
- optional courses Theoretical Computer Science 6 credits
- optional courses Programming 6 credits
- optional courses Computer Sciences 30 credits

Note: Every programme, including the choice of optional courses, has to be discussed and agreed upon with the master coordinator or a personal mentor and approved by the Examination Board.

Vakken:
Doel vak
The objective is to obtain a good understanding of modal logic and its use in computer science and artificial intelligence.

Inhoud vak
A thorough introduction to modal logics, and its applications in computer science and artificial intelligence. We will select some themes from the book Modal Logics for Open Minds, by Johan van Benthem: basic modal logic and possible world semantics, bisimulation and invariance, modal definability, decidability, ... In particular we treat the modal logics most relevant to computer science and AI: temporal, dynamic and epistemic logic.

Onderwijsvorm
Weekly 2 lectures and 1 exercise class, for the duration of 7 weeks.

Toetsvorm
A written exam and assignments that can make half a point bonus.

Literatuur
Johan van Benthem, Modal Logics for Open Minds, CSLI Publications 2010.

Aanbevolen voorkennis
The bachelor course Logica en Modelleren (previously Inleiding Logica), or an equivalent introduction to first-order logic.

Doelgroep
Advanced Selforganisation

Course objective
To understand, simulate and analyse the behaviour and self-organization of complex systems. The student is able to explain, implement and recognize basic principles and properties of such systems.

Course content
This course is about the understanding of the behavior and self-organization of complex systems: systems in which the interaction of the components is not simply reducible to the properties of the components. The general question the we address is: how should systems of very many independent computational (e.g. robotic or software) agents cooperate in order to process information and achieve their goals, in a way that is efficient, self-optimizing, adaptive, and robust in the face of damage or attack? We will look at natural systems that solve some of the same problems that we want to solve, e.g. adaptive path minimization by ants, wasp and termite nest building, army ant raiding, fish schooling and bird flocking, coordinated cooperation in slime molds, synchronized firefly flashing, evolution by natural selection, game theory and the evolution of cooperation. The course includes a practical part in which students implement a simulation of a self-organizing complex system and conduct structured experimental analysis with this simulation.

Form of tuition
Theory in lectures and practice in labs.

Type of assessment
Report including description of simulation and experimental analysis.

Course reading

Target group
mAI-CIS, mAI-HA, mAI-KTIIA, mAI-TAI, mBA, mBA-D, mCS-TAI, mPDCS

Remarks
More information available on BlackBoard. This is a project- oriented course and therefore students will be expected to have basic programming skills.

Advanced Topics in Computer and Network Security
Course objective
The goal of this course is for students to develop an in-depth understanding of classical and recent research in system and network security, and practice their presentation and argumentation skills. The class is restricted only to PDCS students so that individual guidance can be offered.

Course content
The course takes the form of seminars based on a selection of papers that either have had a strong impact on security today, or explore novel ideas that may be important in the future. Students are required to read all papers assigned during the semester and be able to competently discuss the material in class. Each student will be responsible for presenting one lecture -- that lecture will be based on the assigned paper for the week including as much relevant related work as necessary to distill the work presented in the paper. The speaker will have 25 minutes talk to present the papers she read. The presentation will be followed by 20 minutes of interactive discussion in the class. Before each lecture each student must submit to me at least two thought-provoking questions on the main paper for that week. These questions should critically evaluate the paper (eg, questioning the assumptions). At the end of the semester, each student must write a 4-pages long position paper about one of the topics that has been discussed in class. This can be about the topic the student has presented, or about any other topic that has been discussed in class. This is intended to be an interactive class, and as such, class participation will play a significant role in the grading criteria. Students will be graded on the presentation and analysis of their assigned paper, their participation in discussions and questions.

Form of tuition
Seminar.

Type of assessment
Presentations, participation at seminar, and a 4-page position paper. You will be graded with respect to your presentation, your position paper, and your active participation to the seminar. Each of these aspects will account for 1/3 of the final grade. Important: you have to get at least 6 in all the 3 aspects to be able to pass the exam.

Course reading
A selection of papers.

Entry requirements
This course is intended only for students from the masters degree in Parallel and Distributed Computer Systems. Students following other
curricula cannot participate.

**Target group**
This course is only accessible for mPDCS students.

**Remarks**
Course registration is compulsory and must also be done on the first day of lecture directly with the lecturer.

### Advanced Topics in Distributed Systems

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_405022 (405022)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 1</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>dr. G.E.O. Pierre</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>dr. G.E.O. Pierre</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

**Course objective**
Discuss advanced topics relevant for traditional and modern distributed systems.

**Course content**
The course takes the form of a seminar that is based on a selection of papers that either have had a strong impact on distributed systems today, or explore novel ideas that may be important in the future. Subjects will cover important aspects of distributed systems such as communication, data consistency, replication, fault tolerance, performance, scalability, etc. Also, modern distributed systems such as next-generation Web-based systems and wireless sensor networks will have their place. For this seminar we expect the students to actively participate by means of presentations and discussions. Papers for discussions will be selected from the base set, with possibly 1 or 2 added where appropriate.

**Form of tuition**
Seminar.

**Type of assessment**
Presentations, participation at seminar, and a 4-page position paper.

**Course reading**
A (selection of a) list of papers, yet to be decided.

**Entry requirements**
Distributed Systems (400130).

**Target group**
mPDCS

**Remarks**
This course is only accessible for mPDCS students.
More information about this course is available at
http://www.cs.vu.nl/~gpierre/courses/atds/

Cluster and Grid Computing

<table>
<thead>
<tr>
<th>Course code</th>
<th>X 400362 (400362)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 4</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>dr. ing. T. Kielmann</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>dr. ing. T. Kielmann</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

Course objective
Students shall both explore the area of Cluster and Grid Computing and
develop their skills in critical assessment of scientific literature.

Course content
Both Cluster and Grid computing are areas of rapid technical
developments. Many technical developments are still in flux. We
investigate resource management and scheduling, remote data access,
network and other performance issues, as well as software architecture
and programming models for grids.

Form of tuition
Introductory lecture, followed by a seminar part and practical
programming assignments. In the seminar part, students explore topic
areas of Cluster and Grid Computing in small groups, present their
findings in a presentation session and prepare a report. The practical
programming assignments are to be addressed individually.

Type of assessment
Both parts contribute 50% to the grade:
(i) seminar presentation and report
(ii) programming assignments

Course reading
Various research articles as available online.

Entry requirements
Parallel Programming (code 400161)

Target group
mPDCS, mCS-HPDC

Remarks
Participation in the course is limited; priority is given to students
of the M. Sc. programme in Parallel and Distributed Computer Systems,
and to students following the HPDC specialization of; the Msc in
Computer Science. Registration for the course is required before the
first lecture by sending email to the lecturer; first come first serve.

Coding and Cryptography
Doel vak
To give an introduction the theory of error correcting codes and to cryptography.

Inhoud vak
This course provides a thorough introduction to the theory of error correcting codes, and to cryptography. It is aimed especially at students of Computer Science. For error correcting codes we shall include cyclic codes, BCH codes, Reed-Solomon codes and burst error correction. For cryptography we discuss some modern public key cryptography (e.g., RSA, ElGamal, DSA).

Onderwijsvorm
Lectures and exercise classes

Toetsvorm
Written exam and homework

Literatuur
We shall be working from "Coding theory and cryptography, the essentials" by Hankerson, Hoffman, Leonard, Lindner, Phelps, Rodger and Wall (second edition, revised and expanded).

Aanbevolen voorkennis
"Algebra en Discrete Wiskunde 1", "Inleiding Codering en Discrete Wiskunde", or equivalent.

Doelgroep
mAI-CIS, mAI-TAI, mCS-FMSV, mCS-HPDC, mCS-IWT, mCS-MM, mCS-SE, mCS-TAI, mMath, mPDCS

Computer and Network Security

<table>
<thead>
<tr>
<th>Course code</th>
<th>X. 400127 (400127)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 5</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>prof. dr. ir. H.J. Bos</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>prof. dr. ir. H.J. Bos</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

Course objective
Introductory course on security with a scope that includes systems work. At the end of the course students will understand the basic
notion of memory corruption attacks (buffer overflows, format strings, etc), SQL injection, cross-site scripting attacks, and other vectors used by computer hackers. Also, they will be able to understand and apply cryptography.

**Course content**
The course covers a wide spectrum of security issues. We explicitly aim wider than cryptography, as we want to show students how hackers penetrate systems. Part of the course will be hands-on: in lab assignments, student will carry out and investigate attacks in a controlled environment. This involves programming at the both the highest and lowest levels (say SQL and assembly). However, we will also discuss cryptography and trust infrastructures.

**Form of tuition**
Lectures and practical assignments

**Type of assessment**
Written exam (50%) and practical assignments (50%).

**Course reading**
No set book. All material will be made available during the course.

**Entry requirements**
No formal requirements, except a keen interest. Programming experience in C strongly recommended.

**Target group**
mCS, mPDCS

**Remarks**
http://www.few.vu.nl/~herbertb/sec/

**Computer Graphics**

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_400106 (400106)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 2</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>dr. ing. T. Kielmann</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>dr. ing. T. Kielmann</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

**Course objective**
The students shall get theoretical insights and practical knowledge that allows them to implement graphics applications and to understand how such applications get executed on current graphics hardware.

**Course content**
The course has a top-down structure, starting with the applications. Topics of the lecture are:
- graphics programming with OpenGL
- color, input, interaction
- transformations (translation, rotation, scaling, shear)
- 3-dimensional viewing (projections, perspective)
- light and shading
- discrete techniques, buffers, texture mapping
- modeling (object hierarchies, scene graphs)
- advanced topics (curves and surfaces, programmable shaders)

**Form of tuition**
Class sessions consist of lecture elements and practical exercises. Towards the end of the course, an individual programming project will be done by which students apply what they have learned throughout the class. All programming exercises and the project will be based on OpenGL and the programming language Java. It is highly recommended that students bring their own laptop computer to the class sessions in order to perform the practical exercises.

**Type of assessment**
Written exam (code 4001061) and programming project (code 4001062). The project contributes 2/3 to the final grade. The exam contributes 1/3. Both parts need to be graded sufficient or better in order to get the credit points for the course 400106.

**Course reading**

**Entry requirements**
Project Programmeren (400559)

**Target group**
mCS, mPDCS

**Remarks**
The classes are given in period 2; the programming project is continued in period 3.
Onderwijsvorm
Practical computer work

Toetsvorm
Practical computer work.

Aanbevolen voorkennis
Mandatory: Computer Networks (400487).
Strongly recommended: Operating Systems (400011)

Doelgroep
mAI-TAI, mCS-FMSV, mCS-HPDC, mCS-IWT, mPDCS

Overige informatie
Assignments can be submitted until the end of August. Students who have
not taken the Operating Systems course at the VU are strongly advised to
do so, or to study and experiment with the corresponding material
themselves.

Concurrency and Multithreading

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_405064 (405064)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Periode 5</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Faculteit</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coördinator</td>
<td>prof. dr. W.J. Fokkink</td>
</tr>
<tr>
<td>Docent(en)</td>
<td>prof. dr. W.J. Fokkink</td>
</tr>
<tr>
<td>Lesmethode(n)</td>
<td>Hoorcollege, Werkcollege</td>
</tr>
</tbody>
</table>

Doel vak
This course provides a comprehensive presentation of the foundations
and programming principles for multicore machines.

Inhoud vak
Shared memory, mutual exclusion, synchronization operations, concurrent
data structures, scheduling, transactional memory, multithreaded
programming.

Onderwijsvorm
Lectures: 4 hours per week, exercise classes: 4 hours per week.

Toetsvorm
Written exam (which counts for 70% of the final mark) and two
programming assignments (which together count for 30% of the final
mark).

Literatuur
Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming,
Morgan Kaufmann, 2008.

Doelgroep
mAI-CIS, mAI-KTIIA, mAI-TAI, mCS-FMSV, mCS-HPDC, mCS-IWT, mCS-MM, mCS-
SE, mCS-TAI, mPDCS
Overige informatie
The lectures and written exam of the BSc and MSc variant of Concurrency & Multithreading coincide.
The difference is that the BSc variant has two small programming assignments, while the MSc variant has one small and one large programming assignment.

The MSc variant of this course cannot be followed by students that included the BSc variant in their BSc program.

Data Mining Techniques

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_400108 (400108)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 5</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>dr. Z. Szlavik</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>dr. Z. Szlavik</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

Course objective
The aim of the course is that students acquire data mining knowledge and skills that they can apply in a business environment.
How the aims are to be achieved: Students will acquire knowledge and skills mainly through the following: an overview of the most common data mining algorithms and techniques (in lectures), a survey of typical and interesting data mining applications, and practical assignments to gain "hands on" experience. The application of skills in a business environment will be simulated through various assignments of the course.

Course content
The course will provide a survey of basic data mining techniques and their applications for solving real life problems. After a general introduction to Data Mining we will discuss some "classical" algorithms like Naive Bayes, Decision Trees, Association Rules, etc., and some recently discovered methods such as boosting, Support Vector Machines, and co-learning. A number of successful applications of data mining will also be discussed: marketing, fraud detection, text and Web mining, possibly bioinformatics. In addition to lectures, there will be an extensive practical part, where students will experiment with various data mining algorithms and data sets. The grade for the course will be based on these practical assignments (i.e., there will be no final examination).

Form of tuition
Lectures and compulsory practical work. Lectures are planned to be interactive: there will be small questions, one-minute discussions, following an algorithm on paper, looking for patterns in a dataset about you (!), filling in missing pieces in a table, coming up with a number of creative solutions to a small problem, etc.
**Type of assessment**
Practical assignments (i.e. there is no exam). There will be three assignments, some (parts) of these will be done individually, some in groups of two. There is a possibility to get a grade without doing these assignments: one (!) group can be selected (based on interviews conducted by the lecturer) to do a real research project instead (which - be warned - will most likely to involve more work, but it can also be more rewarding).

**Course reading**

**Entry requirements**
Kansrekening en Statistiek of Algemene Statistiek (knowledge of statistics and probabilities) or equivalent. Recommended: Machine Learning.

**Target group**
mBMI, mCS, mAI, mBio

---

### Distributed Algorithms

<table>
<thead>
<tr>
<th>Course code</th>
<th>X. 400211 (400211)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 4</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>prof. dr. W.J. Fokkink</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>prof. dr. W.J. Fokkink</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture, Seminar</td>
</tr>
</tbody>
</table>

**Course objective**
To obtain a good understanding of concurrency concepts and a large range of distributed algorithms.

**Course content**
Snapshots, graph traversal, termination detection, garbage collection, deadlock detection, routing, election, minimal spanning trees, anonymous networks, fault tolerance, failure detection, synchronization, consensus, mutual exclusion, self-stabilization, on-line scheduling.

**Form of tuition**
4 hours per week HC
4 hours per week WC

**Type of assessment**
Written examen (plus a take-home exercise sheet that can provide up to 0,5 bonus point).

**Course reading**
Lecture notes by the lecturer.
Distributed Systems

Course objective
After taking this course, you will be able to:

- understand to a large extent the intricacies related to designing and developing a distributed computer system.

- understand the tradeoffs between centralized, distributed, and fully decentralized solutions.

- be capable of successfully studying research papers on (advanced) distributed systems.

Course content
It is difficult to imagine a standalone modern computer system: every such system is one way or the other connected through a communication network with other computer systems. A collection of networked computer systems is generally referred to as a distributed (computer) system. As with any computer system, we expect a distributed system to simply work, and often even behave as if it were a single computer system. In other words, we would generally like to see all the issues related to the fact that data, processes, and control are actually distributed across a network hidden behind well-defined and properly implemented interfaces. Unfortunately, life is not that easy.

As it turns out, distributed systems time and again exhibit emergent behavior that is difficult to understand by simply looking at individual components. In fact, many aspects of a distributed system cannot even be confined to a few components, as is easily seen by just considering security.

In this course, we pay attention to the pillars on which modern distributed systems are built. Unfortunately, these pillars cannot be viewed independently from each other: each one is equally important.
for understanding why a distributed system behaves the way it does, and depends on the way that other pillars have been constructed. In this sense, pillars form principles, in turn offering a view that one can take when studying distributed systems. We will consider the following principles:

- architectures
- processes
- communication
- naming
- coordination
- consistency and replication
- fault tolerance
- security

These principles will be discussed in the context of a few simplifying concepts that have been used to master the complexity of developing distributed systems: objects, files, documents, and events.

**Form of tuition**
The course is taught as a series of lectures.

**Type of assessment**
There is a written exam.

**Course reading**

**Entry requirements**
Students should have taken a standard course on computer networks. Experience with (distributed) programming will be helpful.

**Target group**
mcS, PDCS, mAI, mIS

---

**Evolutionary Computing**

<table>
<thead>
<tr>
<th>Course code</th>
<th>X. 400111 (400111)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 1</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>prof. dr. A.E. Eiben</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>prof. dr. A.E. Eiben</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

**Course objective**
To learn about computational methods based on Darwinian principles of evolution. To illustrate the usage of such methods as problem solvers and as simulation, respectively modelling tools. To gain hands-on experience in performing experiments.

**Course content**
The course is treating various algorithms based on the Darwinian evolution theory. Driven by natural selection (survival of the fittest), an evolution process is being emulated and solutions for a given problem are being "bred". During this course all "dialects" within evolutionary computing are treated (genetic algorithms, evolution strategies, evolutionary programming, genetic programming, and classifier systems). Applications in optimisation, constraint handling and machine learning are discussed. Specific subjects handled include: various genetic structures (representations), selection techniques, sexual and asexual genetic operators, (self-)adaptivity. If time permits, subjects in Artificial Life and Artificial Societies, and Evolutionary Art will be handled. Hands-on-experience is gained by a compulsory programming assignment.

**Form of tuition**
Oral lectures and compulsory programming assignment.

**Type of assessment**
Written exam and programming assignment (weighted average).

**Course reading**

**Target group**
mBA, mAI, mCS, mPDCS

**Industrial Internship**

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_405080 ()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Ac. Jaar (september)</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Faculteit</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coördinator</td>
<td>dr. G.E.O. Pierre</td>
</tr>
</tbody>
</table>

**Doel vak**
Deepening insights by applying study contents in an industrial setting

**Inhoud vak**
Individual project work by which the student applies the study contents in an industrial setting. Before the start of the internship, the student has to get approval for the internship project by a VU (Computer Science) lecturer. The project has to focus on research or development aspects, by which the student can apply and validate the study contents within the specific constraints of an industrial setting. At the end of the internship, the student submits a written report to the lecturer, in which the work, the lessons learned, and the insights from applying study contents in an industrial setting are described.

**Onderwijsvorm**
Individual project work in an industrial setting

**Toetsvorm**
written report

Aanbevolen voorkennis
36 credit points within the programme mPDCS or mCS

Doelgroep
mCS-FMSV, mCS-HPDC, mCS-IWT, mCS-MM, mCS-SE, mCS-TAI, mPDCS

Internet programming

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_405082 ()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Periode 1</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Faculteit</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coördinator</td>
<td>dr. S. Voulgaris</td>
</tr>
<tr>
<td>Docent(en)</td>
<td>dr. S. Voulgaris</td>
</tr>
<tr>
<td>Lesmethode(n)</td>
<td>Hoorcollege</td>
</tr>
</tbody>
</table>

Doel vak
Guide the student through the design and development of Network and Web applications.

Inhoud vak
The course discusses the principles for understanding, designing, and developing Internet applications. This includes programming the network (sockets, threads, RPC, RMI), programming the web interface (servlets, PHP, Javascript, AJAX), and setting up secure communication channels. Throughout the course, as well as in the context of the lab assignments, attention is paid to practical issues of applying these concepts.

Onderwijsvorm
Lectures combined with lab assignments

Toetsvorm
Final exam plus lab assignments

Literatuur
Course slides

Vereiste voorkennis
Knowledge of C, Java

Aanbevolen voorkennis
Systems Programming (X_400377)
preferred: Computer Networks, Distributed Systems

Doelgroep
mAI-CIS, mAI-CS, mAI-KTIIA, mAI-TAI, mCS-FMSV, mCS-HPDC, mCS-IWT, mCS-MM, mCS-SE, mCS-TAI, mPDCS

Logical Verification

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_400115 (400115)</th>
</tr>
</thead>
</table>
**Course objective**
Introduction to the proof assistant Coq and its foundations

**Course content**
A proof-assistant is used to check the correctness of a specification of a program or the proof of a theorem. The course is concerned with the proof-assistant Coq which is based on typed lambda calculus. In the practical work, we learn to use Coq. One of the exercises is concerned with the correctness proof of the specification of a sorting algorithm, from which a functional program is extracted. In the course, we focus on the Curry-Howard-De Bruijn isomorphism between proofs on the one hand and lambda-terms (which can be seen as functional programs) on the other hand. This is the basis of proof-assistants like Coq. We study various typed lambda calculi and the corresponding logics.

**Form of tuition**
2 hours theory class, 2 hours practical work

**Type of assessment**
Written exam and obligatory Coq exercises

**Course reading**
Course notes

**Entry requirements**
An introduction course in logic.

**Target group**
mCS, mAI, mMath

**Master Project**

<table>
<thead>
<tr>
<th>Course code</th>
<th>X 400461 (400461)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Ac. Year (September)</td>
</tr>
<tr>
<td>Credits</td>
<td>36.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
</tbody>
</table>

**Course objective**
With the Master project, the student is to demonstrate the ability to integrate knowledge, insights, and skills gained so far in the Master programme, and to apply them to a new or otherwise unknown subject.

**Course content**
The student is offered a research question that needs to be answered following a systematic approach. This approach includes steps such as exploring relevant literature, and will, in general, consist of setting
up and carrying out experiments by means of simulations, emulations, or actual systems software. The results and findings will be described in a thesis conforming to the academic standards in the field. An oral presentation of the project results concludes the project.

**Type of assessment**
The grade will be determined based on written thesis and oral presentation.

**Target group**
mPDCS

**Remarks**
Various lecturers

**Operating Systems**

<table>
<thead>
<tr>
<th><strong>Vakcode</strong></th>
<th>X. 405067 (405067)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Periode</strong></td>
<td>Periode 4</td>
</tr>
<tr>
<td><strong>Credits</strong></td>
<td>6.0</td>
</tr>
<tr>
<td><strong>Voertaal</strong></td>
<td>Engels</td>
</tr>
<tr>
<td><strong>Faculteit</strong></td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td><strong>Coördinator</strong></td>
<td>dr. M.R. Rieback</td>
</tr>
<tr>
<td><strong>Lesmethode(n)</strong></td>
<td>Hoorcollege</td>
</tr>
</tbody>
</table>

**Doel vak**
Get an insight into the internals of operating systems for single-processor machines.

**Inhoud vak**
This course gives an introduction to the internals of Operating Systems. The following topics are covered: processes, synchronization, memory allocation, multi-programming, input / output, protection. The course focuses on the example of the MINIX system.

**Onderwijsvorm**
Lectures

**Toetsvorm**
Written exam

**Literatuur**

**Aanbevolen voorkennis**
Recommended: Computer Systems (X_401030)

**Overige informatie**
Students who have included the course Bedrijfssystemen (400011) in their Bachelor's program must not follow this course for their Master's program.

**Operating Systems Practical**
Doel vak
Gain practical experience with the contents of the Operating Systems course.

Inhoud vak
This practical is divided into two separate exercises, each adding functionality to a different part of the MINIX 3 operating system. Both exercises are required to pass the class. The first exercise adds kernel functionality (process profiling), the second one adds file system functionality (defragmentation). Together, these courses cover a significant part of the content of the Operating Systems course.

Onderwijsvorm
Practical computer work

Toetsvorm
Practical computer work.

Aanbevolen voorkennis
Mandatory: Operating Systems (400011)

Doelgroep
mCS-FMSV, mCS-HPDC, mCS-IWT, mPDCS

Overige informatie
Assignments can be submitted until the end of August, although the assistants may not be available during this two-month grace period. The course is based on MINIX version 3.1.0, that is, the version described in the third edition of the "Operating Systems Design and Implementation" book by Tanenbaum and Woodhull. Students who have not taken the "bedrijfssystemen" or "operating systems" course at the VU are strongly advised to follow the Operating Systems course or study the internals of MINIX 3 in detail before the start of the course. All information about the course is available online at http://www.cs.vu.nl/~bs/.

Parallel Programming
**Onderwijsvorm**
Class with separate practicum (6 ECTS).

**Toetsvorm**
Written exam.

**Literatuur**
To be announced.

**Vereiste voorkennis**
Knowledge about the first part of the class Parallel Programming is recommended (introduction into parallel programming, MPI, and Java).

**Parallel Programming Practical**

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_400162 (400162)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Periode 2+3</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Faculteit</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coördinator</td>
<td>prof. dr. ir. H.E. Bal</td>
</tr>
</tbody>
</table>

**Inhoud vak**
With this practicum, several parallel programs have to be written, using different programming environments, including Java, MPI, and CUDA/OpenCL (for GPUs). The programs must be tested on a parallel machine of the faculty (see http://www.cs.vu.nl/das3) and the performance (speedups) of the programs must be measured, analyzed, and, whenever necessary, optimized. A brief report must be written that explains the approach and discusses the measurements.

**Onderwijsvorm**
Practical computer work.

**Toetsvorm**
Practical computer work.

**Vereiste voorkennis**
Parallel Programming Course.

**Aanbevolen voorkennis**
Parallel Programming Course.

**Doelgroep**
mAI, mCS, mPDCS

**PDCS Programming Project**

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_405054 (405054)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Ac. Year (September)</td>
</tr>
</tbody>
</table>
Course content
PDCS programming projects can be taken instead of other practical courses. They are often related to existing research programs in computer systems. There is no set course description as each project is negotiated individually with the professor supervising and grading it. The assignment aims to offer students challenging projects that are often research-oriented by nature. Students are strongly advised to talk to staff members individually to see whether they have a project that matches the student's interest, as well as the capacity to supervise such a project.

Target group
mPDCS

Remarks
Various lecturers

Performance Analysis of Communication Networks

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_400165 (400165)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Periode 1+2</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Coördinator</td>
<td>prof. dr. R.D. van der Mei</td>
</tr>
<tr>
<td>Docent(en)</td>
<td>dr. ing. T. Kielmann, prof. dr. R.D. van der Mei</td>
</tr>
<tr>
<td>Lesmethode(n)</td>
<td>Hoorcollege</td>
</tr>
</tbody>
</table>

Doel vak
The student will acquire basic knowledge of:
- quantitative models for predicting and analyzing the performance of communication networks;
- traffic models;
- traffic management techniques;
- performance evaluation and approximation techniques,
- performance measurement techniques.
The student will gain experience in the development and analysis of performance models and will learn how to tackle practical performance problems arising in the telecommunications industry.

Inhoud vak
Over the past few years the use of communication services (WWW, mobile voice telephony, mobile Internet access, PC banking, on- line ticket reservation, on- line games, peer-to-peer applications, video services) has experienced tremendous growth. Consequently, communication networks are expected to handle huge amounts of (digital) information, and in many situations the available amounts of transmission or processing capacity is a limiting factor, which in many cases leads to degradation of the Quality of Service (QoS). A key factor for the commercial success of communication services in the
competitive telecommunications market is the ability to deliver a high and predictable QoS level to the customers (in terms of response times, throughput and availability) in a cost-effective manner. Typical questions that will be addressed during the course are:
- What does the traffic in the network look like?
- How can we measure performance of the network?
- How many customers can a given network handle with good quality?
- How can we predict the performance of a service in the network?
- How do we deal with traffic problems in the network?
In addition to the basic theory of performance models for communication networks, the application of the theory to solve practical problems will play a central role.

Onderwijsvorm
The course is 2 hours per week. Practical homework assignments will be distributed.

Doelgroep
mBA, mCS, mPDCS, mEct

Overige informatie
This course might be helpful for master students Mathematics.

Research Proposal Writing

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_405023 (405023)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Period 2</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>prof. dr. A.S. Tanenbaum</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>prof. dr. A.S. Tanenbaum</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

Course objective
The aim of this course is to learn to read papers at a research level, organize the material for the framework of a seminar talk, and practice presentation skills for such talk. At the end of this course, the student will be able to acquire the prerequisites for reading and understanding a paper by researching the literature on his own, understand the logic of a paper, and be able to critically evaluate a paper. He or she will be able to extract and condense the material for a talk of a fixed length, and give a captivating and interesting talk to fellow students.

Course content
This course has the single main aim to teach students the first steps of writing a research proposal. Students are asked to evaluate a number of existing proposals that were submitted by staff members in recent years. In addition, each student will also have to write his or her own proposal, which is then evaluated by fellow students following a procedure very similar to what happens in real life. This class may only be attended by PDCS students.
Target group
mPDCS

Scientific Writing in English

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X. 400592 (400592)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Ac. Jaar (september), Periode 4, Periode 5, Periode 6</td>
</tr>
<tr>
<td>Credits</td>
<td>3.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Faculteit</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Lesmethode(n)</td>
<td>Hoorcollege, Werkcollege</td>
</tr>
</tbody>
</table>

**Doel vak**
The aim of this course is to provide the writing student with the essential linguistic means for producing English academic texts which are effective, idiomatically and stylistically appropriate and grammatically correct.

**Inhoud vak**
The initial focus in the course lies on the form of scientific texts in the Exact Sciences:
- Abstract (or summary)
- Introduction
- Methods
- Results
- Discussion

General course outline
Introducing the topics
- Academic and technical writing in English
- The characteristics of different kinds of scientific texts
- How scientific writing is judged and assessed
- Where do you find your information and how do you present it?
- How to avoid committing plagiarism
Who am I writing for? What do I want to say?
- Your readership
- Key parts of an academic article: title, abstract, introduction, methods, results and discussion
Writing the actual article
- Paragraph and sentence construction: how do I link paragraphs together?
- Writing simple and complex sentences. Active and passive sentences.
- Argumentation: how do I put an argument? How do I frame my own opinion?
Should I use “I” or “we”?
Writing correct English
- Use of apostrophes and colons
- Word order, verb tenses, time and tense
- Avoiding mistakes typically made by Dutch writers
- Common spelling mistakes

You will be making considerable use of peer assessment: examining fellow students’ written work and giving them feedback. This method provides useful insights into how a text might be improved. The process of providing someone else with feedback on their text is something that you will find very instructive.
Onderwijsvorm
The course is focused on self-tuition. The plenary sessions concentrate on the process of writing and the product of writing. Homework is part of the course. With each topic, participants work through a phased series of exercises that usually conclude with the requirement to write a short piece of text. The instructor will append extensive written remarks to this text.

Toetsvorm
There will be no examination. However, students will receive their credits only when they have participated in all classes (presence is obligatory) and also when they have handed in the assignments satisfactorily. Students will receive a ‘pass’ when they have finished the course.

Literatuur
The reader ‘Writing a Scientific Article’ can be obtained at the Taalcen- trum-VU in the Metropolitan (4th floor). The costs are 20 euro.

Vereiste voorkennis
Bachelor Exact Sciences

Doelgroep
Compulsory for mAI, mCS, mMNS, mMATH, mBMI & mSFM.
Optional for mIS, mBIO, mPDCS, mCh, mDDS, mPhys.

Overige informatie
Registration for this course is only possible at the Educational Office of the Faculty of Exact Sciences. The usual registration deadlines are applicable, except for period 1. Registration for period 1 is possible until the 24th of August 2012.
Opening hours and contact information can be found here: http://www.few.vu.nl/en/about-faculty/faculty-services/educational-office/index.asp

Period 1 Optional: mAI, mCH, mCS, mDDS, mIS, mPDCS
Period 2 Optional: mAI, mCH, mCS, mDDS, mIS, mPDCS
Period 3 Compulsary: mMNS
Period 3 Optional: mAI, mCS, mCh, mDDS, mIS, mPDCS, mPhys-AMEP, mPhys-PLH, mPhys-PPAP
Period 4 Compulsary: mMath, mBA, mSFM
Period 4 Optional: mAI, mCH, mCS, mDDS, mIS, mPDCS, mPhys-AMEP, mPhys-PLH, mPhys-PPAP
Period 5 Optional: mAI, mCH, mCS, mDDS, mIS, mPDCS, mPhys-AMEP, mPhys-PLH, mPhys-PPAP
Period 6 Optional: mAI, mCH, mCS, mDDS, mIS, mPDCS, mPhys-PLH, mPhys-PPAP

Co-ordinator: drs J.K.A. Meijer

Selected Topics in Parallel and Distributed Computer Systems

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_400426 (400426)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Ac. Jaar (september)</td>
</tr>
</tbody>
</table>
Doel vak
The goal of this course is to treat special topics, which are otherwise not included in the regular curriculum, to individual students as part of further preparation for their masters degree.

Inhoud vak
The actual content of the course is to be decided after consultation of one the staff members, who will act as project supervisor.

Onderwijsvorm
Self study.

Toetsvorm
To be decided by the supervisor.
This course can also be extended to 6 cp (X_400379).

Literatuur
To be decided by the supervisor.

Vereiste voorkennis
None specific.

Doelgroep
mPDCS

Selected Topics in PDCS

<table>
<thead>
<tr>
<th>Vakcode</th>
<th>X_400379 (400379)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periode</td>
<td>Ac. Jaar (september)</td>
</tr>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Voertaal</td>
<td>Engels</td>
</tr>
<tr>
<td>Faculteit</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
</tbody>
</table>

Doel vak
The goal of this course is to treat special topics, which are otherwise not included in the regular curriculum, to individual students as part of further preparation for their masters degree.

Inhoud vak
The actual content of the course is to be decided after consultation of one the PDCS staff members, who will act as project supervisor.

Onderwijsvorm
Self study.

Toetsvorm
To be decided by the supervisor.

Literatuur
To be decided by the supervisor.
Vereiste voorkennis
none specific

Doelgroep
mPDCS

Term Rewriting Systems

<table>
<thead>
<tr>
<th>Course code</th>
<th>X_400121 (400121)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credits</td>
<td>6.0</td>
</tr>
<tr>
<td>Language of tuition</td>
<td>English</td>
</tr>
<tr>
<td>Faculty</td>
<td>Faculteit der Exacte Wetenschappen</td>
</tr>
<tr>
<td>Coordinator</td>
<td>dr. R.C. de Vrijer</td>
</tr>
<tr>
<td>Teaching staff</td>
<td>drs. J. Endrullis</td>
</tr>
<tr>
<td>Teaching method(s)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

Course objective
Learning the fundamental notions of term rewriting and getting acquainted with some more advanced topics in the field

Course content
Term rewriting systems (TRSs) provide for a natural formalism for specifying rules of computation and investigating their properties. TRSs are of basic importance for functional programming and for the implementation of abstract data types. Applications can also be found in theorem proving, proof checking and logic programming. Some topics that will be covered in the course are:
- abstract reduction systems;
- critical pairs and Knuth-Bendix completion;
- orthogonality and reduction strategies;
- termination (rpo’s, monotone algebras);
- combinatory logic;
- decidability issues;
- infinitary rewriting.

Form of tuition
Lectures and practice sessions.

Type of assessment
Written examination.

Course reading
Course notes will be provided.

Entry requirements
Compulsory: Inleiding logica.
Advised: Inleiding theoretische informatica.

Target group
mCS, mAI, mMath