Parallel and Distributed Computer Systems MSc

VU University Amsterdam - Faculteit der Exacte Wetenschappen - M Parallel and Distributed Computer Syst - 2015-2016

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 at the FEW-website.

Index

Limited offered course

This course is taught this year only at our University. Philippe Kruchten is Professor of Software Engineering at the University of British Columbia in Vancouver. He is world-famous as the chief designer of the Rational Unified Process (RUP) and currently he is doing research on Agile Architectures. He will give this course specially for our Master Computer Science and Master Information Sciences students.

Courses:

Name	Period	Credits	Code
Watson Innovation	Period 2	6.0	X_405129

Constrained choice Foundations of Computing and Concurrency (6EC)

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.

Courses:

Name	Period	Credits	Code
Coding and Cryptography	Period 1	6.0	X_405041
Concurrency and Multithreading	Period 1	6.0	X_405064
Concurrency Theory	Period 1	6.0	X_418103
Distributed Algorithms	Period 2	6.0	X_400211
Logical Verification		6.0	X_400115
Protocol Validation	Period 5	6.0	X_400117

Constrained Choice Programming (6EC)

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.

Courses:

Name	Period	Credits	Code
Computer Networks Practical	Period 5+6	6.0	X_405072
Operating Systems Practical	Ac. Year (September)	6.0	X_405071

Parallel Programming Practical	Period 2+3	6.0	X_400162
PDCS Programming Project	Ac. Year (September)	12.0	X_405054

Optional Courses

Courses:

Name	Period	Credits	Code
Advanced Logic	Period 4	6.0	X_405048
Binary and Malware Analysis	Period 1	6.0	X_405100
Evolutionary Computing	Period 1	6.0	X_400111
Green Lab	Period 1	6.0	X_418158
Industrial Internship	Ac. Year (September)	6.0	X_405080
Internet programming	Period 1	6.0	X_405082
Performance of Networked Systems	Period 4	6.0	X_405105
Selected Topics in Parallel and Distributed Computer Systems	Ac. Year (September)	3.0	X_400426
Selected Topics in PDCS	Ac. Year (September)	6.0	X_400379
Term Rewriting Systems	Period 2	6.0	X_400121

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.

Courses:

Name	Period	Credits	Code
Advanced Topics in Computer and Network Security	Period 2	6.0	X_405021
Advanced Topics in Distributed Systems	Period 1	6.0	X_405022
Distributed Systems	Period 2	6.0	X_400130
Large-Scale Computing Infrastructures	Period 5	6.0	X_405106
Master Project	Ac. Year (September)	36.0	X_400461
Parallel Programming for High-performance Applications	Period 1	6.0	X_400161
Research Proposal Writing	Period 2	6.0	X_405023
Systems Security	Period 4	6.0	X_405108

Advanced Logic

Course code	X_405048 (405048)
Period	Period 4
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. F. van Raamsdonk
Examinator	dr. F. van Raamsdonk
Teaching staff	dr. F. van Raamsdonk
Teaching method(s)	Lecture, Seminar
Level	500

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

Course content
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.

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

Type of assessment
A written exam and assignments that can make half a point bonus.

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

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

Target group
mAI, mCS, mPDCS

Advanced Topics in Computer and Network Security

Course code	X_405021 (405021)
Period	Period 2
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. B. Crispo
Examinator	dr. B. Crispo

Teaching method(s)	Lecture
Level	600

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, active participation

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.

Target group
This course is restricted to mPDCS students so that individual guidance can be offered.

Advanced Topics in Distributed Systems

Course code	X_405022 (405022)
Period	Period 1
Credits	6.0

Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	J. Urbani
Examinator	J. Urbani
Teaching staff	J. Urbani
Teaching method(s)	Lecture
Level	600

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/

Binary and Malware Analysis

Course code	X_405100 ()
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	C. Giuffrida

Examinator	prof. dr. ir. H.J. Bos
Teaching staff	prof. dr. ir. H.J. Bos, C. Giuffrida
Teaching method(s)	Lecture
Level	600

Course objective
Deepening insights in static and dynamic analysis, applied to binaries and malware

Course content
Binaries in general, and malware in particular, are very hard to analyse. Unlike with source code, you have no idea what the binary does, or even what the data structures look like - let alone what they mean!. Security analysts, forensic experts, and reverse engineers often have to dig their way through such programs to figure out what the code is all about, and where the interesting pieces of information are.

How do they do this? What techniques and tools can they fall back on, and, conversely, what techniques do the malware authors use to prevent this?

This is a (tough) hands-on specialisation course for a small group of motivated students, who will learn essential analysis techniques and methods in both static and dynamic analysis. Not only will they pick apart real malware, they will also be working on a set of cool and very complicated challenges to find a secret burried deep inside a binary program.

For static analysis, we will look in depth at the generation of control flow graphs, and complications that may arise due to indirect calls and jumps (as well as deliberate obfuscation). For dynamic analysis, we will look at data and control flow tracking (dynamic information flow tracking)

Binary patching will be used to circumvent the binary's defenses. To do so, students need to know details about popular binary formats (ELF, PE, etc.), and work with all manner of state-of-art system tools to analyse the binaries (think IDA Pro, OllyDbg, taint analysis tools, etc.).

In addition, students will be exposed to programs that actively fight static and dynamic analysis.

Form of tuition
Hoorcollege and practical

Course reading
Slides and online material

Target group
mCS-HPDC, mCS-IWT, mPDCS

Coding and Cryptography

Course code	X_405041 (405041)
Period	Period 1
Credits	6.0

Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. R.M.H. de Jeu
Examinator	prof. dr. R.M.H. de Jeu
Teaching staff	prof. dr. R.M.H. de Jeu
Teaching method(s)	Lecture
Level	500

Course objective
To give an introduction the theory of error correcting codes and to cryptography.

Course content
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).

Form of tuition
Lectures and exercise classes

Type of assessment
Written exam and homework. The written exam will count for 80 percent of the grade, the homework will count for 20 percent of the grade. If not both the written exam and the homework are at least 55 percent each, then the maximum score will be 54 percent (which constitutes a fail).

Course reading
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).

Recommended background knowledge
Some knowledge on linear algebra, on the integers modulo n, and on polynomials.

Target group
mAI, mCS, mMath, mPDCS

Computer Networks Practical

Course code	X_405072 (405072)
Period	Period 5+6
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. S. Voulgaris
Examinator	dr. S. Voulgaris
Teaching method(s)	Lecture
Level	500

Course objective
Put concepts of Computer Networks and Operating Systems into practice, in the context of smartphones.

Course content
This is a (tough) lab course, that involves low-level programming on Android smartphones. It requires very thorough understanding of operating systems and network concepts. It is done either individually or in groups of two.

Form of tuition
Practical computer work

Type of assessment
Practical computer work

Recommended background knowledge
Computer Networks (400487)
Operating Systems (400011)
Good knowledge of Java!

Target group
mCS, mPDCS

Concurrency and Multithreading

Course code	X_405064 (405064)
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Teaching staff	prof. dr. W.J. Fokkink
Teaching method(s)	Lecture, Seminar
Level	400

Course objective
This course provides a comprehensive presentation of the foundations and programming principles for multicore computing devices.

Learning objectives are: fundamental insight into multicore computing; algorithms for multicore computing; analyzing such algorithms; concurrent datastructures; multicore programming.

Course content
Shared memory, mutual exclusion, synchronization operations, concurrent data structures, scheduling, transactional memory, multithreaded programming assignment.

Form of tuition
Lectures: 4 hours per week, exercise classes: 4 hours per week.

Type of assessment
Written exam (which counts for 70% of the final mark) and one programming assignment (which counts for 30% of the final mark).

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

Target group
mAI, mCS, mPDCS

Remarks
The homepage of the course is at http://www.cs.vu.nl/~tcs/cm/

The lectures and written exam of the BSc and MSc variant of Concurrency and Multithreading coincide. The difference is that the BSc variant has a smaller programming assignment than the MSc variant.

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

Concurrency Theory

Course code	X_418103 ()
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	O.W. Schrofer
Examinator	O.W. Schrofer
Teaching staff	dr. A. Ponse
Teaching method(s)	Lecture, Seminar, Computer lab
Level	600

Course content
http://studiegids.uva.nl/xmlpages/page/2015-2016/zoek-vak/vak/16283

Remarks
This course is offered at the UvA. For more information contact: FNWI Education Service Centre, Science Park 904, servicedesk-esc-science@uva.nl, +31 (0)20 525 7100.
Enrolment via https://m.sis.uva.nl/vakaanmelden is required.

Distributed Algorithms

Course code	X_400211 (400211)
Period	Period 2
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. W.J. Fokkink

Examinator	prof. dr. W.J. Fokkink
Teaching staff	prof. dr. W.J. Fokkink
Teaching method(s)	Lecture, Seminar
Level	500

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

To offer a bird's-eye view on a wide range of algorithms for basic challenges in distributed systems.

To provide students with an algorithmic frame of mind for solving fundamental problems in distributed computing.

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.

Characteristic of the course is that correctness arguments and complexity calculations of distributed algorithms are provided in an intuitive fashion.

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
W.J Fokkink. Distributed Algorithms: An Intuitive Approach. MIT Press, 2013.

Target group
mAI, mCS, mPDCS

Remarks
The homepage of the course is at http://www.cs.vu.nl/~tcs/da/

Distributed Systems

Course code	X_400130 (400130)
Period	Period 2
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Teaching staff	dr. ing. T. Kielmann
Teaching method(s)	Lecture, Seminar

Level	400

Course objective
After taking this course, students 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

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, in combination with small exercises.

Type of assessment
Written exam.

Course reading
This year, we will use a reader. Details about its distribution will be announced via blackboard in due time.

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

Target group
mCS, mPDCS, mAI

Evolutionary Computing

Course code	X_400111 (400111)
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. A.E. Eiben
Examinator	prof. dr. A.E. Eiben
Teaching staff	prof. dr. A.E. Eiben, J.V. Heinerman MSc
Teaching method(s)	Lecture
Level	400

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, evolutiestrategieën, evolutionary programming, genetic programming, and classifier systems). Applications in optimisation, constraint handling, machine learning, and robotics are discussed. Specific subjects handled include:
various genetic structures (representations), selection techniques, sexual and asexual variation operators, (self-)adaptivity. Special attention is paid to methodological aspects, such as algorithm design and tuning. If time
permits, subjects in Artificial Life will be handled. Hands-on-experience is gained by a
compulsory programming assignment.

Form of tuition
Oral lectures and compulsory programming assignment. Highly motivated students can replace the programming assignment by a special research track under the personal supervision of the lecturer(s).

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

Course reading
Eiben, A.E., Smith, J.E., Introduction to Evolutionary Computing.
Springer, 2003 ISBN 3-540-40184-9.
Slides available from http://www.cs.vu.nl/~gusz/ecbook/ecbook.html .

Target group
mBA, mAI, mCS, mPDCS

Green Lab

Course code	X_418158 ()
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. P. Lago
Examinator	prof. dr. P. Lago
Teaching staff	dr. ing. G. Procaccianti
Teaching method(s)	Lecture, Practical
Level	400

Course objective
Learn the basics of empirical experimentation in the field of Software Engineering.
Be able to operate in a lab environment and build a successful experiment for software energy consumption.
Become familiar with the research problems in the field of green software engineering.
Understand and measure the impact of software over energy consumption.

Course content
Students will work in teams to perform experiments on software energy consumption in a controlled environment. They will have to carry out all the phases of empirical experimentation, from experiment design to operation, data analysis and reporting. They will be provided with examples of previous experiments, but they will have to choose by themselves the experimental subjects and hypotheses to test. During the lab sessions, students will be assisted for technical operation of the lab equipment as regards measurement and data gathering. Students will also receive the required training for data analysis and visualization (i.e. graphs, dashboards) using specialized software.

Form of tuition
Lectures. Lab sessions.

Type of assessment
Teamwork, project assignments.

Course reading
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., &
Wesslén, A. (2012). Experimentation in software engineering. Springer.

Material distributed on Blackboard.

Recommended background knowledge
Basic statistical analysis techniques (descriptive statistics and most common tests).

Target group
mCS, PDCS, mAI

Industrial Internship

Course code	X_405080 ()
Period	Ac. Year (September)
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Level	500

Course objective
Deepening insights by applying study contents in an industrial setting

Course content
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.

For the grading of the report, most important are the student's reflections
on study contents vs. "industrial reality": What did you learn during your
studies that was particularly helpful for your internship? What is different
in an industrial environment, compared to university? What did you learn during your internship that you were not told at university?

Form of tuition
individual project work in an industrial setting

Type of assessment
written report

Recommended background knowledge
The student should have completed at least 48 credits of his or her Master programme such that there are sufficient study contents to be applied in an industrial setting.

Target group
mCS, mPDCS

Remarks
Various lecturers

Internet programming

Course code	X_405082 ()
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. S. Voulgaris
Examinator	dr. S. Voulgaris
Teaching staff	dr. S. Voulgaris
Teaching method(s)	Lecture
Level	500

Course objective
Guide the student through the design and development of Network and Web applications.

Course content
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.

Form of tuition
Lectures combined with lab assignments

Type of assessment
Final exam plus lab assignments

Course reading
Course slides

Entry requirements
Knowledge of C, Java

Recommended background knowledge
Good knowledge of both C and Java

Target group
mAI, mCS, mPDCS

Large-Scale Computing Infrastructures

Course code	X_405106 ()
Period	Period 5
Credits	6.0

Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Teaching staff	dr. ing. T. Kielmann
Teaching method(s)	Lecture
Level	500

Course objective
Students explore the field of large-scale computing infrastructures. They study its technological frontiers from scientific publications and get hands-on experience via programming assignments.

Course content
Cloud infrastructures like Amazon's EC2 or Microsoft's Azure provide seemingly limitless compute and storage capacity. The technology underlying these systems strongly relies on decades of work on high-performance, distributed computing platforms, such as cluster computing, computing grids, and supercomputers. We study aspects of computing in large scale, such as resource management and scheduling, remote data access, energy efficiency, failure resilience, performance of large systems, as well as suitable software architecture and programming models such as MapReduce.

Form of tuition
Introductory lectures, followed by a seminar part and practical programming assignments. In the seminar part, students explore the technological frontiers of large-scale computing, published in top-quality scientific venues of the field. Students present their findings and write position papers about topics presented by other students in the class. With the practical programming assignments, students get hands-on experience with large-scale computing infrastructures.

Type of assessment
Both parts contribute 50% each to the grade:

- seminar presentation and position paper

- programming assignments

Course reading
Various scientific articles as available online

Recommended background knowledge
Students should have basic knowledge about distributed systems and parallel application programming.
Students must be able to program in Java and Python (or be able to get the needed skills on the fly).

Target group
mPDCS, mCS

Logical Verification

Course code	X_400115 (400115)

Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. F. van Raamsdonk
Examinator	dr. F. van Raamsdonk
Teaching staff	dr. F. van Raamsdonk
Teaching method(s)	Lecture, Practical
Level	500

Course objective
Introduction to the proof assistant Coq and its type-theoretic 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 times 2 hours theory class, 2 times 2 hours practical work

Type of assessment
Written exam, obligatory Coq-exercises, obligatory hand-in theory exercises.

Course reading
Course notes

Entry requirements
An introduction course in logic.

Target group
mCS, mAI, mMath, mPDCS

Remarks
The course is taught once every two years, the next opportunity will be in study year 2016-2017

Master Project

Course code	X_400461 (400461)
Period	Ac. Year (September)
Credits	36.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann

Examinator	dr. ing. T. Kielmann
Level	600

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 the quality of the performed project work, a written thesis, and an oral presentation.

Recommended background knowledge
The student must have completed (almost) the complete study program before starting the Master Project.

Target group
mPDCS

Remarks
Various lecturers

Operating Systems Practical

Course code	X_405071 (405071)
Period	Ac. Year (September)
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	ir. M.P.H. Huntjens
Examinator	ir. M.P.H. Huntjens
Teaching staff	ir. M.P.H. Huntjens
Teaching method(s)	Lecture
Level	500

Course objective
Gain practical experience with the contents of the Computer Networks and Operating Systems courses.

Course content
Three problems covering (as much as possible) the content of the courses.

Form of tuition
Practical computer work

Type of assessment
Practical computer work.

Target group
mCS, mPDCS

Remarks
The course is not taught in 2014-2015, the next opportunity will be probably in 2015-2016

Parallel Programming for High-performance Applications

Course code	X_400161 (400161)
Period	Period 1
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. ir. H.E. Bal
Examinator	prof. dr. ir. H.E. Bal
Teaching staff	prof. dr. ir. H.E. Bal
Teaching method(s)	Lecture
Level	400

Course objective
You will learn
- the backgrounds of High Performance Computing (HPC)
- apply design methods for parallel algorithms
- compare different parallel computer architectures
- analyze performance of network topologies
- compare different parallel programming constructs and programming environments
- get insight in some selected parallel applications
- understand many-cores (e.g. GPUs), many-core algorithms, many-core optimizations

Course content
This lecture discusses how programs can be written that run in parallel on a large number of processors, with the goal of reducing execution time. The class has a brief introduction into parallel computing systems (architectures). The focus of the class, however, is on programming methods, languages, and applications. Both traditional techniques (like message passing) and more advanced techniques (like parallel object-oriented languages) will be discussed. Several parallel applications are discussed, including N-body simulations and search algorithms. About 4 lectures are devoted to an important new development: programming many-core machines such as Graphical Processing Units (GPUs). The class fits well with existing research projects within the department of Computing Systems. It is a good basis for M.Sc. projects in the area of parallel programming, which use the parallel computing systems of the department.

Form of tuition

Lectures (4 hours per week), given by prof.dr.ir. Henri Bal (VU) and dr. Ana Varbanescu (UvA). There is a separate Parallel Programming Practicum (6 ECTS).

Type of assessment
Written exam.

Course reading
To be announced.

Target group
mAI, mBIO, mCS, mPDCS, m Computational Science

Parallel Programming Practical

Course code	X_400162 (400162)
Period	Period 2+3
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. ir. H.E. Bal
Examinator	prof. dr. ir. H.E. Bal
Teaching staff	prof. dr. ir. H.E. Bal
Teaching method(s)	Lecture
Level	500

Course content
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/das4) 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.

Form of tuition
Practical computer work.

Type of assessment
Practical computer work.

Entry requirements
Knowledge of parallel programming in Java, MPI, and CUDA/OpenCL (as taught in the Parallel Programming course) is required.

Target group
Masters Computer Science, PDCS, AI, and Computational Science

Remarks
Students can do this course either in Period 2 or in Period 3.
It is not possible to submit assignments in both periods.

Lecturers:
prof. dr. ir. H.E. Bal
Dr. C. Grelck

PDCS Programming Project

Course code	X_405054 (405054)
Period	Ac. Year (September)
Credits	12.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Level	600

Course objective
Let the student get hands-on experience with developing systems-level software in the context of ongoing research projects

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.

Form of tuition
Individual programming project

Type of assessment
To be decided by the supervisor.

Course reading
To be decided by the supervisor.

Target group
mPDCS

Remarks
Various lecturers

Performance of Networked Systems

Course code	X_405105 ()
Period	Period 4
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen

Coordinator	prof. dr. R.D. van der Mei
Examinator	prof. dr. R.D. van der Mei
Teaching staff	dr. ing. T. Kielmann, prof. dr. R.D. van der Mei
Teaching method(s)	Lecture
Level	400

Course objective
Students will acquire basic knowledge of:
• performance aspects of networked systems, consisting of servers, services, and clients
• performance engineering principles and methods,
• quantitative models for predicting and optimizing the performance of networked systems,
• quantitative models for planning capacity of networked systems.
Students will gain experience in engineering and planning performance of networked systems, and will learn how to tackle practical performance problems arising in the ICT industry.

Course content
Over the past few decades, information and communication technology (ICT) has become ubiquitous and globally interconnected. As a consequence, our information and communication systems are expected to process huge amounts of (digital) information, which puts a tremendous burden on our ICT infrastructure. At the same time, our modern society has become largely dependent on the well-functioning of our ICT systems; large-scale system failures and perceivable Quality of Service (QoS) degradation may completely disrupt our daily lives and have huge impact on our economy.

Motivated by this, the course will focus on performance-related issues of networked systems. In the first part, we study capacity planning and modeling for server systems and networks. In the second part, we study the client side of performance while focusing on web applications for both desktop and mobile devices. We address questions like:

- How can we design and engineer networked systems for performance?
- How can we plan server capacity in networked systems?
- How can web applications improve performance across wired and wireless networks?

Form of tuition
Classroom lectures and practical homework assignments.

Type of assessment
The assessment will be based on both homework assignments and a written exam.

Course reading
Textbook, supplemented with a reader on Stochastic Performance Modelling.

High Performance Browser Networking, Ilya Grigorik, O-Reilly, 2013.

Entry requirements
The students should have basic knowledge of computer networks.

Target group
mBA, mCS, mPDCS, mEct

Protocol Validation

Course code	X_400117 (400117)
Period	Period 5
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Teaching method(s)	Lecture, Practical, Seminar
Level	500

Course objective
Learning to use formal techniques for specification and validation of communication protocols.

Course content
This course is concerned with the specification and validation of protocols, using formal methods. The course is based on a specification language based on process algebra combined with abstract data types, called mCRL. This language and its toolset can be used for the specification of parallel, communicating processes with data. Model checking is a method for expressing properties of concurrent finite-state systems, which can be checked automatically. Interesting properties of a specification are: "something bad will never happen" (safety), and "something good will eventually happen" (liveness). In the lab we will teach the use of a tool for automated verification of the required properties of a specification.

Form of tuition
4 hours per week HC
4 hours per week WC/PR (mixed)

During the practicum the mCRL tool and the CADP model checker will be used for the validation of protocols discussed during lectures.

Type of assessment
Written exam, together with a practical homework assignment. The overall mark of the course is (H+2W)/3, where H is the mark for the homework assignment, and W is the mark for the written exam.

Course reading
Wan Fokkink, Modelling Distributed Systems, Springer 2007.

Recommended background knowledge
Logica en Modelleren

Target group
mAI, mCS, mPDCS

Remarks

The course is taught once every two years, the next opportunity will be in study year 2015-2016

Research Proposal Writing

Course code	X_405023 (405023)
Period	Period 2
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. ir. H.E. Bal
Examinator	prof. dr. ir. H.E. Bal
Teaching staff	prof. dr. ir. H.E. Bal
Teaching method(s)	Lecture
Level	600

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

Selected Topics in Parallel and Distributed Computer Systems

Course code	X_400426 (400426)
Period	Ac. Year (September)
Credits	3.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Level	500

Course objective
The goal of this course is to treat special topics, in the field of parallel and distributed computer systems, that are otherwise not included in the regular curriculum, to individual students as part of further preparation for their master's degree.

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

Form of tuition
Individual study under guidance of the supervisor.

Type of assessment
To be decided by the supervisor.

Course reading
To be decided by the supervisor.

Entry requirements
None specific.

Target group
mPDCS

Remarks
Two versions of this course exist. X_400426 has a volume of 3EC. X_400379 has a volume of 6EC.

Selected Topics in PDCS

Course code	X_400379 (400379)
Period	Ac. Year (September)
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Level	500

Course objective
The goal of this course is to treat special topics, in the field of parallel and distributed computer systems, that are otherwise not included in the regular curriculum, to individual students as part of further preparation for their master's degree.

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

Form of tuition
Individual study under guidance of the supervisor.

Type of assessment

To be decided by the supervisor.

Course reading
To be decided by the supervisor.

Entry requirements
none specific

Target group
mPDCS

Remarks
Two versions of this course exist. X_400426 has a volume of 3EC.
X_400379 has a volume of 6EC.

Systems Security

Course code	X_405108 ()
Period	Period 4
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	prof. dr. ir. H.J. Bos
Examinator	prof. dr. ir. H.J. Bos
Teaching staff	prof. dr. ir. H.J. Bos
Teaching method(s)	Lecture, Seminar
Level	500

Course objective
Very tough course on security with a focus on systems work. At the end of the course students will deeply understand the basic notion of memory corruption attacks (buffer overflows, format strings, etc), network attacks, SQL injection, cross-site scripting attacks, and other vectors used by computer hackers. Besides basic attack, students will learn about state-of-the-art exploitation methods. The course is very(!) hands-on.

Course content
The course covers a wide spectrum of security issues. We explicitly focus on systems security rather than (say) cryptography, as we want to show students how attackers penetrate systems.

Specifically, the course focuses on (1) network security (sniffing, spoofing, hijacking, exploiting network protocols, DDoS, DNS attacks, etc.), (2) memory corruption and application security (buffer overflows, format string bugs, dangling pointers, shellcode, return oriented programming, ASLR/DEP/canaries, control flow integrity and cool new ways of exploitation), (3) web security (XSS, SQL injection, CSRF, http cache poisoning, SOP, authentication, etc.), (4) botnets (centralised/P2P, fast flux, double flux), (4) crypto (basics, systems aspects).

Much of the course will be hands-on and challenge-based. In 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).

Form of tuition
Lectures and (very challenging) practical assignments.

Type of assessment
Written exam (30%) and practical assignments (70%).

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

Entry requirements
Knowledge of C is probably essential

Recommended background knowledge
No formal requirements, except a keen interest and a lot of time.
Programming experience
in C very strongly recommended. Knowledge of assembly and computer
architecture helps too.

Target group
mCS, mPDCS.

Term Rewriting Systems

Course code	X_400121 (400121)
Period	Period 2
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	drs. J. Endrullis
Examinator	drs. J. Endrullis
Teaching staff	drs. J. Endrullis
Teaching method(s)	Lecture, Seminar
Level	600

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

Target group
mCS, mPDCS, mAI, mMath

Remarks
The course is taught once every two years, the next opportunity will be in study year 2016-2017

Watson Innovation

Course code	X_405129 ()
Period	Period 2
Credits	6.0
Language of tuition	English
Faculty	Faculteit der Exacte Wetenschappen
Coordinator	dr. L.M. Aroyo
Examinator	dr. L.M. Aroyo
Teaching staff	dr. L.M. Aroyo, A. Dumitrache MSc, B.F.L. Timmermans MSc
Teaching method(s)	Lecture
Level	400

Course objective
The Watson Innovation course is a collaboration between VU Amsterdam and IBM. In this course you will learn the basics and challenges of Cognitive Computing and how to train Cognitive Computing Systems. You will have the unique opportunity to work with multidisciplinary teams on real prototypes of IBM Watson, and explore its potential for answering questions about the city of Amsterdam. You will also have a chance to showcase developed applications and plans to real clients.

Course content
- Basics of Cognitive Computing & IBM Watson
- How to train IBM Watson Instance
- Develop ideas for Cognitive Computing apps
- Build real IBM Watson prototype apps
- Showcase your ideas to real clients

Form of tuition
Lectures & practical sessions at locations of the VU Amsterdam and IBM Netherlands.

Type of assessment
Evaluation of group projects and individual peer-reviews

Course reading

Course lecture slides and related articles:
- What is IBM Watson?
(http://www.ibm.com/smarterplanet/us/en/ibmwatson/what-is-watson.html)
- Building Watson: An overview of the DeepQA project
(http://www.aaai.org/ojs/index.php/aimagazine/article/download/2303/2165
)
- CrowdTruth papers (http://crowdtruth.org/papers/)

Target group
A balanced mix of Computer Science and Business & Economics students (from VU as well as UvA) in their bachelor or master level.

Registration procedure
Sign up through VUnet and http://crowdtruth.org/course.
For more information contact b.timmermans@vu.nl.

Places are limited, so sign up as soon as possible.

Remarks
There will be no lectures through the Christmas period. The period from 18 December till 10 January is reserved for students individual and group work. Office hours will be provided for additional feedback and questions.