Computer Science MSc

Vrije Universiteit Amsterdam - Faculteit der Exacte Wetenschappen - M Computer Science - 2016-2017

Computer Science studies the technology that has become ubiquitous in our global, connected society. Traditionally, the computer had been the primary object of study. Nowadays, globally distributed information processing services have taken center stage, with the Internet connecting a wide variety of information processing devices, ranging from mobile phones to data centers operated by the world leadership companies.

The technical side of Computer Science deals with computer operations, like system software, computer networks, and programming environments. The theoretical foundations cover, for example, limits of what can be computed, computational efficiency, correctness, and the intricacies of concurrent execution. Software engineering studies construction and maintenance of large and often mission-critical software systems that need to be maintained by large teams of people. Because of its prominent role in everyday life, non-functional aspects of information technology are gaining importance in Computer Science, most notably the energy efficiency of data centers, and the security of computer systems.

Students aiming to enroll in the Master Computer Science are required to have a solid background on the level of a Bachelor of Science in Computer Science, or similar. The program is structured in multiple, focused specializations from which a student chooses one according to personal preferences. The program is organized in close collaboration with the University of Amsterdam.
From the academic year 2016 on, students will be awarded a joint degree from both universities. Classes are taught both at the VU campus, and at Science Park Amsterdam. Both locations are conveniently connected by both public transportation and bicycle paths.

Inhoudsopgave

Track Big Data Engineering

Opleidingsdelen:

- Constrained choice Foundations of Computing and Concurrency (6 EC)
- Constrained Choice Mathematics
- Constrained choice Software Engineering (6 EC)
- Pre-approved Elective courses BDE
- Compulsory Courses
- Mastercore

Constrained choice Foundations of Computing and Concurrency (6 EC)

Compulsory choice Theoretical Computer Science at least 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Distributed Algorithms	Periode 1	6.0	X_400211
Logical Verification	Periode 5	6.0	X_400115
Protocol Validation		6.0	X_400117
Term Rewriting Systems	Periode 4+5	6.0	XM_400121

Constrained Choice Mathematics

Compulsory choice of at least one Mathematics course of 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Coding and Cryptography	Periode 1	6.0	X_405041
Experimental Design and Data Analysis	Periode 5	6.0	X_405078

Constrained choice Software Engineering (6 EC)

Each student must follow at least 6EC from the core track courses of
Track Software Engineering

Vakken:

Naam	Periode	Credits	Code
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170

Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439

Pre-approved Elective courses BDE

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Binary and Malware Analysis	Periode 5	6.0	X_405100
Business Process Analytics	Periode 4	6.0	X_400650
Business Process Management	Periode 1	6.0	X_405115
Computer and Network Security	Periode 1	6.0	X_400127
Data Mining Techniques	Periode 5	6.0	X_400108
Developing Services for the Cloud	Periode 3	6.0	X_405074
Distributed Algorithms	Periode 1	6.0	X_400211
Evolutionary Computing	Periode 1	6.0	X_400111
Green Lab	Periode 1	6.0	X_418158
High Performance Computing and Big Data	Periode 3	6.0	XMU_40013
ICT4D: Information and communication technology for Development	Periode 5	6.0	X_405101
Industrial Internship	Ac. Jaar (september)	6.0	XM_405080
Information Visualization	Periode 4	6.0	XMU_418143
Internet programming	Periode 1	6.0	X_405082
Introduction to Computational Science	Periode 1	6.0	XMU_418111
Kernel Programming	Periode 1	6.0	XM_40014
Knowledge and Media	Periode 1	6.0	X_405065
Knowledge Engineering	Periode 2+3	6.0	X_405099
Lambda Calculus	Periode 2	6.0	XMU_418108
Large Scale Data Engineering	Periode 1	6.0	X_405116
Logical Verification	Periode 5	6.0	X_400115
Parallel System Architectures	Periode 1	6.0	XMU_40015
Performance Engineering	Periode 5	6.0	XMU_40016
Performance of Networked Systems	Periode 4	6.0	X_405105
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017

Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Protocol Validation		6.0	X_400117
Secure Software	Periode 2	6.0	XM_40019
Serious Games	Periode 5	6.0	X_405097
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439
Term Rewriting Systems	Periode 4+5	6.0	XM_400121
The Social Web	Periode 4	6.0	X_405086
Watson Innovation	Periode 2	6.0	X_405129
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Compulsory Courses

Vakken:

Naam	Periode	Credits	Code
Data Mining Techniques	Periode 5	6.0	X_400108
Information Visualization	Periode 4	6.0	XMU_418143
Large Scale Data Engineering	Periode 1	6.0	X_405116
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Mastercore

Vakken:

Naam	Periode	Credits	Code
Distributed Systems	Periode 2	6.0	X_400130
History of digital cultures	Periode 3	6.0	XMU_418107
Literature Study and Seminar	Ac. Jaar (september)	6.0	X_405111
Master Project	Ac. Jaar (september)	36.0	XM_400442

Track Computer Systems Security

Opleidingsdelen:

- Constrained choice Foundations of Computing and Concurrency (6 EC)
- Constrained Choice Mathematics
- Constrained choice Software Engineering (6 EC)
- Pre-approved Elective courses
- Compulsory Courses
- Mastercore

Constrained choice Foundations of Computing and Concurrency (6 EC)

Compulsory choice Theoretical Computer Science at least 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Distributed Algorithms	Periode 1	6.0	X_400211
Logical Verification	Periode 5	6.0	X_400115
Protocol Validation		6.0	X_400117
Term Rewriting Systems	Periode 4+5	6.0	XM_400121

Constrained Choice Mathematics

Compulsory choice of at least one Mathematics course of 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Coding and Cryptography	Periode 1	6.0	X_405041
Experimental Design and Data Analysis	Periode 5	6.0	X_405078

Constrained choice Software Engineering (6 EC)

Each student must follow at least 6EC from the core track courses of
Track Software Engineering

Vakken:

Naam	Periode	Credits	Code
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412

Software Testing	Periode 4	6.0	X_400439

Pre-approved Elective courses

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Binary and Malware Analysis	Periode 5	6.0	X_405100
Business Process Analytics	Periode 4	6.0	X_400650
Business Process Management	Periode 1	6.0	X_405115
Computer and Network Security	Periode 1	6.0	X_400127
Data Mining Techniques	Periode 5	6.0	X_400108
Developing Services for the Cloud	Periode 3	6.0	X_405074
Distributed Algorithms	Periode 1	6.0	X_400211
Evolutionary Computing	Periode 1	6.0	X_400111
Green Lab	Periode 1	6.0	X_418158
High Performance Computing and Big Data	Periode 3	6.0	XMU_40013
ICT4D: Information and communication technology for Development	Periode 5	6.0	X_405101
Industrial Internship	Ac. Jaar (september)	6.0	XM_405080
Information Visualization	Periode 4	6.0	XMU_418143
Internet programming	Periode 1	6.0	X_405082
Introduction to Computational Science	Periode 1	6.0	XMU_418111
Kernel Programming	Periode 1	6.0	XM_40014
Knowledge and Media	Periode 1	6.0	X_405065
Knowledge Engineering	Periode 2+3	6.0	X_405099
Lambda Calculus	Periode 2	6.0	XMU_418108
Large Scale Data Engineering	Periode 1	6.0	X_405116
Logical Verification	Periode 5	6.0	X_400115
Parallel System Architectures	Periode 1	6.0	XMU_40015
Performance Engineering	Periode 5	6.0	XMU_40016
Performance of Networked Systems	Periode 4	6.0	X_405105
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018

Protocol Validation		6.0	X_400117
Secure Software	Periode 2	6.0	XM_40019
Serious Games	Periode 5	6.0	X_405097
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439
Term Rewriting Systems	Periode 4+5	6.0	XM_400121
The Social Web	Periode 4	6.0	X_405086
Watson Innovation	Periode 2	6.0	X_405129
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Compulsory Courses

Vakken:

Naam	Periode	Credits	Code
Binary and Malware Analysis	Periode 5	6.0	X_405100
Computer and Network Security	Periode 1	6.0	X_400127
Cybercrime and Forensics	Periode 4	6.0	XMU_40012
Kernel Programming	Periode 1	6.0	XM_40014
Secure Software	Periode 2	6.0	XM_40019

Mastercore

Vakken:

Naam	Periode	Credits	Code
Distributed Systems	Periode 2	6.0	X_400130
History of digital cultures	Periode 3	6.0	XMU_418107
Literature Study and Seminar	Ac. Jaar (september)	6.0	X_405111
Master Project	Ac. Jaar (september)	36.0	XM_400442

Track Foundations of Computing and Concurrency

This track aims at Computer Science students with a general interest in the application of formal methods in computing, concurrency and the design and verification of software systems. Some theoretical

disciplines that play a central role are term rewriting, process algebra, distributed algorithms and type theory. Foundational disciplines include logic, recursion theory and complexity.

All these topics have a wide range of applications, of which we mention just a few. Tools developed from process algebra are used in protocol validation. Term rewriting is used in the execution of equational specifications and lies at the basis of functional programming and the analysis of infinitary processes. Distributed algorithms are of central importance for the efficient use of concurrent systems. Logic and type theory form the basis of proof checking, used in software verification.

In all of the above areas courses are offered. To mention a few examples: Distributed Algorithms, Logical Verification, Protocol Validation, Term Rewriting Systems, Concurrent System Design by Abstraction. The programme can be enhanced by choosing one or more appropriate mathematics courses.

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.

Master Coordinator:

dr. F. van Raamsdonk
K room T-446
T +31 (0) 20 598 7710
E f.van.raamsdonk@vu.nl

Opleidingsdelen:

- Constrained Choice Mathematics
- Pre-approved Elective courses
- Constrained Choice Programming
- Constrained Choice Software Engineering (6EC)
- Compulsory Courses
- Mastercore

Constrained Choice Mathematics

Compulsory choice of at least one Mathematics course of 6 credits, recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Coding and Cryptography	Periode 1	6.0	X_405041
Experimental Design and Data Analysis	Periode 5	6.0	X_405078

Pre-approved Elective courses

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Binary and Malware Analysis	Periode 5	6.0	X_405100
Business Process Analytics	Periode 4	6.0	X_400650
Business Process Management	Periode 1	6.0	X_405115
Computer and Network Security	Periode 1	6.0	X_400127
Data Mining Techniques	Periode 5	6.0	X_400108
Developing Services for the Cloud	Periode 3	6.0	X_405074
Distributed Algorithms	Periode 1	6.0	X_400211
Evolutionary Computing	Periode 1	6.0	X_400111
Green Lab	Periode 1	6.0	X_418158
High Performance Computing and Big Data	Periode 3	6.0	XMU_40013
ICT4D: Information and communication technology for Development	Periode 5	6.0	X_405101
Industrial Internship	Ac. Jaar (september)	6.0	XM_405080
Information Visualization	Periode 4	6.0	XMU_418143
Internet programming	Periode 1	6.0	X_405082
Introduction to Computational Science	Periode 1	6.0	XMU_418111
Kernel Programming	Periode 1	6.0	XM_40014
Knowledge and Media	Periode 1	6.0	X_405065
Knowledge Engineering	Periode 2+3	6.0	X_405099
Lambda Calculus	Periode 2	6.0	XMU_418108
Large Scale Data Engineering	Periode 1	6.0	X_405116
Logical Verification	Periode 5	6.0	X_400115
Parallel System Architectures	Periode 1	6.0	XMU_40015
Performance Engineering	Periode 5	6.0	XMU_40016
Performance of Networked Systems	Periode 4	6.0	X_405105
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Protocol Validation		6.0	X_400117
Secure Software	Periode 2	6.0	XM_40019
Serious Games	Periode 5	6.0	X_405097
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412

Software Testing	Periode 4	6.0	X_400439
Term Rewriting Systems	Periode 4+5	6.0	XM_400121
The Social Web	Periode 4	6.0	X_405086
Watson Innovation	Periode 2	6.0	X_405129
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Constrained Choice Programming

Compulsory choice Practical Work Computer Science at least 6 credits, recommended are 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:

Naam	Periode	Credits	Code
Android Lab	Periode 5+6	6.0	XM_40011
Concurrency and Multithreading	Periode 2	6.0	X_405064
Individual Systems Practical	Ac. Jaar (september)	6.0	XM_405088
Internet programming	Periode 1	6.0	X_405082
Kernel Programming	Periode 1	6.0	XM_40014
Parallel Programming Practical	Periode 2+3	6.0	X_400162
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Software Testing Practical	Periode 6	6.0	X_405124

Constrained Choice Software Engineering (6EC)

Vakken:

Naam	Periode	Credits	Code
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439

Compulsory Courses

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Distributed Algorithms	Periode 1	6.0	X_400211
Logical Verification	Periode 5	6.0	X_400115
Protocol Validation		6.0	X_400117
Term Rewriting Systems	Periode 4+5	6.0	XM_400121

Mastercore

Vakken:

Naam	Periode	Credits	Code
Distributed Systems	Periode 2	6.0	X_400130
History of digital cultures	Periode 3	6.0	XMU_418107
Literature Study and Seminar	Ac. Jaar (september)	6.0	X_405111
Master Project	Ac. Jaar (september)	36.0	XM_400442

Track Internet and Web Technology

The Internet and the World Wide Web play an ever more central role in our society. This specialisation is concerned with large-scale computer systems, especially computer networks and the Internet. Important topics are: Internet and Web protocols, distributed systems, network security, development tools for network applications, peer-to-peer technology, etc.

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.

Master Coordinator:

dr. S. Voulgaris
K room R-447
T +31 (0) 20 598 3715
E spyros.voulgaris@vu.nl

Opleidingsdelen:

- Constrained Choice Mathematics
- Pre-approved Elective courses
- Core track courses
- Mastercore

Constrained Choice Mathematics

Compulsory choice of at least one Mathematics course of 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Coding and Cryptography	Periode 1	6.0	X_405041
Experimental Design and Data Analysis	Periode 5	6.0	X_405078

Pre-approved Elective courses

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Binary and Malware Analysis	Periode 5	6.0	X_405100
Business Process Analytics	Periode 4	6.0	X_400650
Business Process Management	Periode 1	6.0	X_405115
Computer and Network Security	Periode 1	6.0	X_400127
Data Mining Techniques	Periode 5	6.0	X_400108
Developing Services for the Cloud	Periode 3	6.0	X_405074
Distributed Algorithms	Periode 1	6.0	X_400211
Evolutionary Computing	Periode 1	6.0	X_400111
Green Lab	Periode 1	6.0	X_418158
High Performance Computing and Big Data	Periode 3	6.0	XMU_40013
ICT4D: Information and communication technology for Development	Periode 5	6.0	X_405101
Industrial Internship	Ac. Jaar (september)	6.0	XM_405080
Information Visualization	Periode 4	6.0	XMU_418143
Internet programming	Periode 1	6.0	X_405082
Introduction to Computational Science	Periode 1	6.0	XMU_418111
Kernel Programming	Periode 1	6.0	XM_40014
Knowledge and Media	Periode 1	6.0	X_405065
Knowledge Engineering	Periode 2+3	6.0	X_405099
Lambda Calculus	Periode 2	6.0	XMU_418108
Large Scale Data Engineering	Periode 1	6.0	X_405116
Logical Verification	Periode 5	6.0	X_400115

Parallel System Architectures	Periode 1	6.0	XMU_40015
Performance Engineering	Periode 5	6.0	XMU_40016
Performance of Networked Systems	Periode 4	6.0	X_405105
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Protocol Validation		6.0	X_400117
Secure Software	Periode 2	6.0	XM_40019
Serious Games	Periode 5	6.0	X_405097
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439
Term Rewriting Systems	Periode 4+5	6.0	XM_400121
The Social Web	Periode 4	6.0	X_405086
Watson Innovation	Periode 2	6.0	X_405129
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Core track courses

Vakken:

Naam	Periode	Credits	Code
Distributed Algorithms	Periode 1	6.0	X_400211
Internet programming	Periode 1	6.0	X_405082
Performance of Networked Systems	Periode 4	6.0	X_405105
Service Oriented Design	Periode 1	6.0	X_405061
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Mastercore

Vakken:

Naam	Periode	Credits	Code
Distributed Systems	Periode 2	6.0	X_400130
History of digital cultures	Periode 3	6.0	XMU_418107

Literature Study and Seminar	Ac. Jaar (september)	6.0	X_405111
Master Project	Ac. Jaar (september)	36.0	XM_400442

Track Parallel Computing Systems

Opleidingsdelen:

- Constrained choice Foundations of Computing and Concurrency (6 EC)
- Constrained Choice Mathematics
- Constrained choice Software Engineering (6 EC)
- Compulsoy Courses
- Pre-approved Elective courses
- Mastercore

Constrained choice Foundations of Computing and Concurrency (6 EC)

Compulsory choice Theoretical Computer Science at least 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Distributed Algorithms	Periode 1	6.0	X_400211
Logical Verification	Periode 5	6.0	X_400115
Protocol Validation		6.0	X_400117
Term Rewriting Systems	Periode 4+5	6.0	XM_400121

Constrained Choice Mathematics

Compulsory choice of at least one Mathematics course of 6 credits,
recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Coding and Cryptography	Periode 1	6.0	X_405041
Experimental Design and Data Analysis	Periode 5	6.0	X_405078

Constrained choice Software Engineering (6 EC)

Each student must follow at least 6EC from the core track courses of
Track Software Engineering

Vakken:

Naam	Periode	Credits	Code
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439

Compulsoy Courses

Vakken:

Naam	Periode	Credits	Code
Parallel Programming Practical	Periode 2+3	6.0	X_400162
Parallel System Architectures	Periode 1	6.0	XMU_40015
Performance Engineering	Periode 5	6.0	XMU_40016
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018

Pre-approved Elective courses

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Binary and Malware Analysis	Periode 5	6.0	X_405100
Business Process Analytics	Periode 4	6.0	X_400650
Business Process Management	Periode 1	6.0	X_405115
Computer and Network Security	Periode 1	6.0	X_400127
Data Mining Techniques	Periode 5	6.0	X_400108
Developing Services for the Cloud	Periode 3	6.0	X_405074
Distributed Algorithms	Periode 1	6.0	X_400211
Evolutionary Computing	Periode 1	6.0	X_400111
Green Lab	Periode 1	6.0	X_418158
High Performance Computing and Big Data	Periode 3	6.0	XMU_40013
ICT4D: Information and communication technology for Development	Periode 5	6.0	X_405101

Industrial Internship	Ac. Jaar (september)	6.0	XM_405080
Information Visualization	Periode 4	6.0	XMU_418143
Internet programming	Periode 1	6.0	X_405082
Introduction to Computational Science	Periode 1	6.0	XMU_418111
Kernel Programming	Periode 1	6.0	XM_40014
Knowledge and Media	Periode 1	6.0	X_405065
Knowledge Engineering	Periode 2+3	6.0	X_405099
Lambda Calculus	Periode 2	6.0	XMU_418108
Large Scale Data Engineering	Periode 1	6.0	X_405116
Logical Verification	Periode 5	6.0	X_400115
Parallel System Architectures	Periode 1	6.0	XMU_40015
Performance Engineering	Periode 5	6.0	XMU_40016
Performance of Networked Systems	Periode 4	6.0	X_405105
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Protocol Validation		6.0	X_400117
Secure Software	Periode 2	6.0	XM_40019
Serious Games	Periode 5	6.0	X_405097
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439
Term Rewriting Systems	Periode 4+5	6.0	XM_400121
The Social Web	Periode 4	6.0	X_405086
Watson Innovation	Periode 2	6.0	X_405129
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Mastercore

Vakken:

Naam	**Periode**	**Credits**	**Code**
Distributed Systems	Periode 2	6.0	X_400130
History of digital cultures	Periode 3	6.0	XMU_418107
Literature Study and Seminar	Ac. Jaar (september)	6.0	X_405111

Master Project	Ac. Jaar (september)	36.0	XM_400442

Track Software Engineering and Green IT

Some people define software engineering as: 'the application of a systematic, quantifiable approach to the development, execution and maintenance of software. It is a broad and comprehensive field, in which engineering plays an important part, next to psychological and managerial aspects. Keywords are evolution and complexity.
The field continually evolves, as the type of systems as well as the world at large changes. New developments such as outsourcing, global system development, service-orientation and the incorporation of off-the-shelf software profoundly influence the field.

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.

Master Coordinator:

Prof.dr. P. Lago
K room T-422
T +31 (0) 20 59 87745
E p.lago@vu.nl

Opleidingsdelen:

- Constrained choice Foundations of Computing and Concurrency (6 EC)
- Constrained Choice Mathematics
- Pre-approved Elective courses
- Compulsory Courses
- Constrained Choice Programming
- Mastercore

Constrained choice Foundations of Computing and Concurrency (6 EC)

Compulsory choice Theoretical Computer Science at least 6 credits, recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Distributed Algorithms	Periode 1	6.0	X_400211
Logical Verification	Periode 5	6.0	X_400115
Protocol Validation		6.0	X_400117
Term Rewriting Systems	Periode 4+5	6.0	XM_400121

Constrained Choice Mathematics

Compulsory choice of at least one Mathematics course of 6 credits, recommended are the courses below.

Vakken:

Naam	Periode	Credits	Code
Coding and Cryptography	Periode 1	6.0	X_405041
Experimental Design and Data Analysis	Periode 5	6.0	X_405078

Pre-approved Elective courses

Vakken:

Naam	Periode	Credits	Code
Advanced Logic	Periode 4	6.0	X_405048
Binary and Malware Analysis	Periode 5	6.0	X_405100
Business Process Analytics	Periode 4	6.0	X_400650
Business Process Management	Periode 1	6.0	X_405115
Computer and Network Security	Periode 1	6.0	X_400127
Data Mining Techniques	Periode 5	6.0	X_400108
Developing Services for the Cloud	Periode 3	6.0	X_405074
Distributed Algorithms	Periode 1	6.0	X_400211
Evolutionary Computing	Periode 1	6.0	X_400111
Green Lab	Periode 1	6.0	X_418158
High Performance Computing and Big Data	Periode 3	6.0	XMU_40013
ICT4D: Information and communication technology for Development	Periode 5	6.0	X_405101
Industrial Internship	Ac. Jaar (september)	6.0	XM_405080
Information Visualization	Periode 4	6.0	XMU_418143
Internet programming	Periode 1	6.0	X_405082
Introduction to Computational Science	Periode 1	6.0	XMU_418111
Kernel Programming	Periode 1	6.0	XM_40014
Knowledge and Media	Periode 1	6.0	X_405065
Knowledge Engineering	Periode 2+3	6.0	X_405099
Lambda Calculus	Periode 2	6.0	XMU_418108
Large Scale Data Engineering	Periode 1	6.0	X_405116
Logical Verification	Periode 5	6.0	X_400115
Parallel System Architectures	Periode 1	6.0	XMU_40015

Performance Engineering	Periode 5	6.0	XMU_40016
Performance of Networked Systems	Periode 4	6.0	X_405105
Programming Large-scale Parallel Systems	Periode 1	6.0	XM_40017
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Protocol Validation		6.0	X_400117
Secure Software	Periode 2	6.0	XM_40019
Serious Games	Periode 5	6.0	X_405097
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439
Term Rewriting Systems	Periode 4+5	6.0	XM_400121
The Social Web	Periode 4	6.0	X_405086
Watson Innovation	Periode 2	6.0	X_405129
Web Data Processing Systems	Periode 2	6.0	XM_40020
Web Services and Cloud-based Systems	Periode 5	6.0	XMU_418110

Compulsory Courses

Vakken:

Naam	Periode	Credits	Code
Green Lab	Periode 1	6.0	X_418158
Service Oriented Design	Periode 1	6.0	X_405061
Software Architecture	Periode 2	6.0	X_400170
Software Asset Management	Periode 1	6.0	X_400412
Software Testing	Periode 4	6.0	X_400439

Constrained Choice Programming

Compulsory choice Practical Work Computer Science at least 6 credits, recommended are 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:

Naam	Periode	Credits	Code

Android Lab	Periode 5+6	6.0	XM_40011
Concurrency and Multithreading	Periode 2	6.0	X_405064
Individual Systems Practical	Ac. Jaar (september)	6.0	XM_405088
Internet programming	Periode 1	6.0	X_405082
Kernel Programming	Periode 1	6.0	XM_40014
Parallel Programming Practical	Periode 2+3	6.0	X_400162
Programming Multi-core and Many-core Systems	Periode 4	6.0	XMU_40018
Software Testing Practical	Periode 6	6.0	X_405124

Mastercore

Vakken:

Naam	Periode	Credits	Code
Distributed Systems	Periode 2	6.0	X_400130
History of digital cultures	Periode 3	6.0	XMU_418107
Literature Study and Seminar	Ac. Jaar (september)	6.0	X_405111
Master Project	Ac. Jaar (september)	36.0	XM_400442

Expired courses

The course modules presented in the list below will no longer be offered in academic year 2016-2017.

Vakken:

Naam	Periode	Credits	Code
Concurrent System Design by Abstraction		6.0	X_418104
Programming Concurrent Systems		6.0	X_418109

Advanced Logic

Vakcode	X_405048 (405048)
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. F. van Raamsdonk
Examinator	dr. F. van Raamsdonk
Docent(en)	dr. F. van Raamsdonk

Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	500

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 Logic and Modelling
or an equivalent introduction to first-order logic.

Doelgroep
mAI, mCS, mPDCS

Android Lab

Vakcode	XM_40011 ()
Periode	Periode 5+6
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. S. Voulgaris
Examinator	dr. S. Voulgaris
Docent(en)	dr. S. Voulgaris
Lesmethode(n)	Hoorcollege, Computerpracticum
Niveau	400

Doel vak
The objective of this practical is to put concepts of Computer Networks and Operating Systems into practice, in the context of Android smartphones. It consists of low-level programming assignments on Android smartphones, requiring the thorough understanding of operating systems and network concepts.

Inhoud vak

This practical consists of two parts. The first is to build a TCP implementation for Android smartphones. The second is to develop a Chat application over the TCP stack of the first part. Both parts are to be written in Java using the Android operating system.

Onderwijsvorm
One project assignment.

Literatuur
Computer Networks, by Andrew S. Tanenbaum

Vereiste voorkennis
Good knowledge of Java.

Aanbevolen voorkennis
Good knowledge of Java.

Doelgroep
mCS, mPDCS

Overige informatie
Lecturer:
dr. S. Voulgaris

Binary and Malware Analysis

Vakcode	X_405100 ()
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	C. Giuffrida
Examinator	C. Giuffrida
Docent(en)	prof. dr. ir. H.J. Bos, C. Giuffrida
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
Deepening insights in static and dynamic analysis, applied to binaries and malware

Inhoud vak
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.

Onderwijsvorm
Hoorcollege and practical

Literatuur
Slides and online material

Doelgroep
mCS-HPDC, mCS-IWT, mPDCS

Business Process Analytics

Vakcode	X_400650 ()
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. H. Leopold MSc
Examinator	dr. H. Leopold MSc
Docent(en)	dr. H. Leopold MSc
Lesmethode(n)	Hoorcollege, Practicum
Niveau	400

Doel vak
After taking this course, the student will:

- be aware of the current possibilities to support BPM with information technology.
- understand and be able to employ process mining techniques for the purpose of process discovery, compliance checking, and improvement.
- know key technologies for analyzing large process model repositories.
- know and be able to employ basic as well as advanced NLP techniques for the purpose of process analysis.
- know and be able to use process model simulation for testing and improving process design.

Inhoud vak
There is a steadily increasing interest of organizations to use Business Process Management (BPM) for documenting and improving their operations. However, the associated manual effort for thoroughly eliciting, documenting, and updating process knowledge in the form of process models is often considerable.

Within this course, we put an emphasis on the technological and analytical perspective and discuss how they can support organizations in effectively and efficiently implementing BPM. In fact, techniques from the fields of information retrieval, data mining as well as simulation provide valuable foundations to reduce to the manual effort in the context of BPM. Hence, we introduce and discuss four different technological angles and demonstrate how each of these angles can strengthen the different phases of the BPM life cycle. In particular, we address the following technological areas:

1. Process Mining: The technology of process mining builds on the analysis of event logs that were generated by information or workflow systems. We discuss how process mining techniques can be used for process discovery, compliance checking, and improvement and elaborate on basic as well as advanced process mining algorithms. In addition, we introduce current process mining tools for the application of process mining in practice.

2. Process Model Collections: Many large organizations maintain process model repositories with several hundred process models. Hence, manual analysis efforts are time-consuming and cumbersome. Recognizing this, we introduce key concepts to automatically analyze process model collections. Among others, we discuss techniques for process model comparison, process model search, and behavioral analysis of process models.

3. Natural Language Analysis: The automated analysis of natural language, which is referred to as Natural Language Processing (NLP), has been applied in many contexts. As an example, consider Apple’s Siri or Google’s S Voice, which are capable of interpreting human speech. In fact, also organizations and their business processes may considerably benefit from natural language processing techniques. Hence, we introduce the key NLP techniques that are relevant in the context of BPM. Among others, we discuss techniques for process model content analysis, process model quality insurance, and identification of improvement potential in process models.

4. Simulation: The simulation of business processes is a tool that is used to predict performance and to understand the impact of change. It, for instance, allows organizations to test processes before they are actually technically implemented in a system. Due to its usefulness for organizations, we introduce the technological foundations for process simulation and give an overview of process simulation tools.

The various lectures and instructions will be devoted to these technological areas.

Onderwijsvorm
There will be lectures as well as work instructions.

Toetsvorm

The grading for students who follow this course in the scheduled period will be based on two grades:

1. The first grade is based on a number of home assignments. The goal of the assignments is to evaluate whether the students can successfully apply the content from the lecture. Among others, the students will be asked to mine a business process model from a given event log and to automatically infer relevant information using natural language processing tools from a given text.

2. The second grade is gained by participating in the regular exam during the exam week. The exam is a closed book exam, which consists of theoretical questions and small assignments. Selected chapters from the books "Fundamentals of Business Process Management", "Process Mining", and "Speech and Language Processing" will be the basis for this exam.

The overall result for this exam is the rounded, weighted average of the first grade (50%) and the second grade (50%) provided that both grades (unrounded) at least amount to a 5.00. If either of the grades is lower than a 5.00, the overall grade for this course is determined by the rounded, lowest grade of the two.

For all students who fail the course in the scheduled period or decide to follow the course outside this period, the course is graded solely by the grade for the re-exam. This is a full exam similar to the original exam and the assignments. The re-exam is a closed book exam, too.

Literatuur
1. Fundamentals of Business Process Management. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. Springer, 2013. ISBN: 978-3-642-33142-8 (Print) 978-3-642-33143-5 (Online).

2. Process Mining. Discovery, Conformance and Enhancement of Business Processes. van der Aalst, Wil. Springer, 2011. ISBN: 978-3642193446.

3. Speech and Language Processing, Jurafsky, Dan, Martin, James H. Pearson International Edition, 2008. ISBN: 978-0135041963.

Aanbevolen voorkennis
Students will, among others, benefit from the knowledge they acquired in the courses Information Management and Business Process Management. Motivated students, however, will be able to master the course without prior knowledge from these courses.

Doelgroep
This is an interdisciplinary course. Any student who is interested in learning how technology can be used to improve business processes in practice is invited to join this course.

Business Process Management

Vakcode	X_405115 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen

Coördinator	prof. dr. ir. H.A. Reijers
Examinator	prof. dr. ir. H.A. Reijers
Docent(en)	prof. dr. ir. H.A. Reijers
Lesmethode(n)	Hoorcollege, Practicum, Werkcollege
Niveau	400

Doel vak
Business Process Management is a rapidly growing field, both in practice and academia. Evidence from the effectiveness of process-oriented approaches is accumulating. Process-aware technologies are used by organizations in all areas of the world, in all sectors.

As an expert in Business Information Systems, it is inevitable that you will get involved in process improvement projects. In your career, you may find yourself in the role of a professional working in a process that is being analyzed, redesigned, or supported by information technology. Alternatively, you may be managing such a process. Even more likely, you may play the role of intermediary, standing between the operational professionals executing a process and higher management that wishes organizational improvement. The knowledge and especially the skills taught in this course provide you with the basic instruments to carry out and understand BPM projects.

This course also gives a view on the scientific challenges that the BPM field is concerned with. This may stimulate you to contribute to the solutions for these challenges, for example as a scientist in this area.

After taking this course, the student will be able to:
• explain the organizational merits of process thinking, in particular in contrast to traditional management thinking;
• identify the different phases in the management of business processes;
• model complex business processes with a formal modeling technique, taking (partly) informal requirements into account;
• communicate process designs to both end-users and IT specialists;
• use process design theory to develop alternatives to existing processes;
• analyze the conformance and performance of process designs before they are put into production;
• understand how business processes can be analyzed on the basis of analyzing event logs;
• describe and understand the main features of process-aware information systems (workflow technology).

Inhoud vak
As a response to increasing competition and more demanding customers, various researchers, practitioners, and management gurus have suggested companies to put less emphasis on hierarchical and functional structures, but instead focus on and improve entire chains of business operations, ranging often from client to client. The orientation on such business processes to manage and improve organizational effectiveness is at the core of this course.
Within this course, there is an emphasis on the role of models and information technology to manage business processes. This means that there will be a focus on the creation and analysis of design artifacts, in particular process models. Also, the role of IT as an enabling and support technology for process improvement will receive a wide share of

attention.
The course on Business Process Management builds on the idea that business processes go through a life-cycle, with different phases:
• Identification: the problem to distinguish which processes in organizations require priority to be actively managed;
• Discovery: the elicitation and specification of the way that operational processes are carried out;
• Analysis: the understanding of a process' structural ability to fulfill the requirements it must meet;
• Redesign: the planned actions to increase the performance and/or conformance of business processes by changing its elements;
• Implementation: the execution of business processes using advanced IT, such as workflow management systems;
• Monitoring/control: the day-to-day monitoring of a business process to detect operational problems and violations of regulations.
The various lectures and instructions will be devoted to these phases.

Onderwijsvorm
Three hours of lectures per week (h) and two hours of work instructions (w).

Toetsvorm
Assignments (O) and a closed-book exam (T). The resit is one integrated, closed-book exam (T).

Literatuur
Fundamentals of Business Process Management. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. Springer, 2013. ISBN: 978-3-642-33142-8 (Print) 978-3-642-33143-5 (Online).

Coding and Cryptography

Vakcode	X_405041 (405041)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. R.M.H. de Jeu
Examinator	prof. dr. R.M.H. de Jeu
Docent(en)	prof. dr. R.M.H. de Jeu
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
To give an introduction the theory of error correcting codes and to discuss some algebraic background of cryptography.

Inhoud vak
This course provides a thorough introduction to the theory of error correcting codes, and also, as a small part of it, treats the algebraic background of some protocols in 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 the small part on cryptography we discuss some modern public key cryptography (e.g., RSA, ElGamal, DSA).

Onderwijsvorm
Lectures and exercise classes

Toetsvorm
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).

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

Doelgroep
XM_CS 1, XM_PDCS 1, XM_MAT_B 1, XM_MAT_E 1, XM_MAT_T 1, XM_MAT_S 1, XM_MAT_ADS 1, XM_MAT_AG 1

Computer and Network Security

Vakcode	X_400127 (400127)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. ir. H.J. Bos
Examinator	prof. dr. ir. H.J. Bos
Docent(en)	prof. dr. ir. H.J. Bos
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
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.

Inhoud vak
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).

Onderwijsvorm
Lectures and (very challenging) practical assignments.

Toetsvorm
Written exam (30%) and practical assignments (70%).

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

Vereiste voorkennis
Knowledge of C is probably essential

Aanbevolen voorkennis
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.

Concurrency and Multithreading

Vakcode	X_405064 (405064)
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Docent(en)	prof. dr. W.J. Fokkink
Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	400

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

Specific learning objectives are:
* To provide insight into fundamental notions of multicore computing and their relation to practice: locks, read-modify-write operations, mutual exclusion, consensus, construction of atomic multi-reader-multi-writer registers, lost wakeups, ABA problem.
* To provide insight into algorithms and frameworks for multicore computing and their application in multi-threaded programs: mutual exclusion algorithms, spin locks, monitors, barriers, AtomicStampedReference class in Java, thread pools in Java, transactional memory.
* Analyzing algoritmes for multicore computing with regard to

functionality and performance: linearizability, starvation- and wait-freeness, Amdahl's law, compute efficiency gain of parallelism.
* Mastering elementary datastructures in the context of multicore computing: lists, queues, stacks.
* Programming in multi-threaded Java, and performing experiments with such programs.

Inhoud vak
The course consists of the following topics: Shared memory, mutual exclusion, synchronization operations, concurrent data structures, scheduling, transactional memory, and a multithreaded programming assignment.

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

Toetsvorm
The written exam counts for 65% and the programming assignment for 35% of the final mark.

Both for the written exam and the programming assignment at least a 5.0 must be obtained (and the overall average mark should be at least 5.5).

Only students that achieved at least a 3.0 for their initial programming assignment are offered a resit opportunity for this assignment.

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

Aanbevolen voorkennis
Datastructures & Algorithms

Programming in Java

Doelgroep
mAI, mCS, mPDCS

Overige informatie
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.

Concurrent System Design by Abstraction

Vakcode	X_418104 ()
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	O.W. Schrofer
Examinator	O.W. Schrofer

Lesmethode(n)	Hoorcollege
Niveau	600

Inhoud vak
This course has expired

Cybercrime and Forensics

Vakcode	XMU_40012 ()
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Niveau	400

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2016-2017/zoek-vak/vak/26305

Data Mining Techniques

Vakcode	X_400108 (400108)
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. M. Hoogendoorn
Examinator	dr. M. Hoogendoorn
Docent(en)	dr. M. Hoogendoorn
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
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.

Inhoud vak
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).

Onderwijsvorm
Lectures (h) and compulsory practical work (pra). Lectures are planned to be interactive: there will be small questions, one-minute discussions, etc.

Toetsvorm
Practical assignments (i.e. there is no exam). There will be two assignments done in groups of three. There is a possibility to get a grade without doing these assignments: to do a real research project instead (which will most likely to involve more work, but it can also be more rewarding). For the regular assignments the first assignment counts for 40% and the second for 60%. The grade of both assignments needs to be sufficient to pass the course.

Literatuur
Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques (Third Edition). Morgan Kaufmann, January 2011
ISBN 978-0-12-374856-0

Aanbevolen voorkennis
Kansrekening and Statistiek or Algemene Statistiek (knowledge of statistics and probabilities) or equivalent. Recommended: Machine Learning.

Doelgroep
mBA, mCS, mAI, mBio

Developing Services for the Cloud

Vakcode	X_405074 ()
Periode	Periode 3
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. P. Lago
Examinator	prof. dr. P. Lago
Docent(en)	prof. dr. P. Lago
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
Learn how to design and implement software services by using modern technologies and development environments; transforming SoaML models into code, experience implementation challenges of software services; troubleshooting in teams; deployment in the cloud.

Inhoud vak
Both service oriented computing and cloud computing are booming areas of rapid development and adoption. This module addresses major related technical aspects including how to transform a service oriented design

into service implementations, how to deploy them in a remote cloud environment, how to effectively and critically use development tools. Group work and active participation is an essential part of the classes. Form of tuition: Lectures and group work.

Onderwijsvorm
H, W, pra, pro.

Toetsvorm
Final presentation with demo. Teamwork.

Literatuur
Material distributed by lecturers.

Vereiste voorkennis
Knowledge of UML and SoaML; software engineering and Java programming.

Aanbevolen voorkennis
Preferred: module Service Oriented Design (405061).

Doelgroep
mCS

Overige informatie
Registration is compulsory at least 4 weeks before course starts.
Attendance is compulsory (all students bring their laptop!).
The students attending the module will be instructed to prepare required input material before the module starts.

Lecturers:
prof. P. Lago
Guest lecturers from IBM

Distributed Algorithms

Vakcode	X_400211 (400211)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Docent(en)	prof. dr. W.J. Fokkink
Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	500

Doel vak
The main objective is to provide students with an algorithmic frame of mind for solving fundamental problems in distributed computing. They obtain insight into concurrency concepts, and are offered a bird's-eye view on a wide range of algorithms for basic and important challenges in distributed systems.

Characteristic of the course is that correctness arguments and

complexity calculations of distributed algorithms are provided in an intuitive fashion and by means of examples and exercises.

Inhoud vak
The following topics are treated in the course: Logical clocks, 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.

Onderwijsvorm
4 hours per week HC
4 hours per week WC

Toetsvorm
Written examen (plus a take-home exercise sheet that can provide up to 0.5 bonus point).

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

Aanbevolen voorkennis
Datastructures & Algorithms

Doelgroep
mAI, mCS, mPDCS

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

Distributed Systems

Vakcode	X_400130 (400130)
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. ing. T. Kielmann
Examinator	dr. ing. T. Kielmann
Docent(en)	dr. ing. T. Kielmann
Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	400

Doel vak
After taking this course, students will be able to:

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

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

- understand the impact of scalability on performance and fault-tolerance of a distributed system.

- understand the impact of performance and fault tolerance on data consistency.

- understand the peculiarities of process coordination on large scale.

Inhoud vak
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 principles from which modern distributed systems are built. Unfortunately, these principles cannot be viewed independently from each other: each one is equally important for understanding why a distributed system behaves the way it does. 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.

Onderwijsvorm
The course is taught as a series of lectures, in combination with exercise classes.

Toetsvorm
Written exam.

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

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

Doelgroep

mCS, mPDCS, mSNE (UvA)

Evolutionary Computing

Vakcode	X_400111 (400111)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. A.E. Eiben
Examinator	prof. dr. A.E. Eiben
Docent(en)	prof. dr. A.E. Eiben, J.V. Heinerman MSc
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
To learn about computational methods based on Darwinian principles of evolution. To illustrate the usage of such methods as problem solvers and as simulation tools.To gain hands-on
experience in performing experiments.

Inhoud vak
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.

Onderwijsvorm
Oral lectures and compulsory programming assignment (in teams of 3). Highly motivated students can replace the programming assignment by a special research track under the personal supervision of the lecturer (s).

Toetsvorm
Written exam and pogramming assignment (weighted average).

Literatuur
Eiben, A.E., Smith, J.E., Introduction to Evolutionary Computing. Springer, 2015, 2nd edition, ISBN 978-3-662-44873-1.

Vereiste voorkennis
Programming skills are necessary to do the practical assignment.

Doelgroep
mBA, mAI, mCS, mPDCS

Experimental Design and Data Analysis

Vakcode	X_405078 ()
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. E.N. Belitser
Examinator	dr. E.N. Belitser
Docent(en)	dr. E.N. Belitser
Lesmethode(n)	Hoorcollege, Practicum
Niveau	400

Doel vak
In this course the student is acquainted with the most common experimental designs and regression models. Furthermore, nonparametric tests and bootstrap methods are discussed. On completion of this course the student should be able to:
- design experiments and analyse the results according to the design,
- analyse data using the common ANOVA designs,
- analyse data using linear regression or a generalized linear regression model,
- perform basic nonparametric tests,
- perform bootstrap and permutation tests.

Inhoud vak
Regression models try to explain or predict a dependent variable using measured independent variables. Statistical methods are needed if there is random variation in the dependent variables. We will discuss multiple linear regression, analyses of variance (ANOVA), generalized linear regression models. All methods will be illustrated with practical examples. Especially in the case of ANOVA it is necessary that the study is well designed in order to draw sound conclusions from an experiment or survey. In this course a few well known designs (completely randomized, randomized block etc.) and the associated analyses of variance are discussed. The remainder of the course is be dedicated to non-parametric testing methods and bootstrap methods:
- Wilcoxon test for (one and two samples),
- Kolmogorov-Smirnov test (two samples),
- rank correlation tests,
- permutation and bootstrap tests.
All analyses are carried out by using the statistical package R.

Onderwijsvorm
Lectures, computer classes, discussions of the computer assignments.

Toetsvorm
Weekly computer assignments and final assignment. The final grade is based on the written reports of all these assignments.

Literatuur
- Slides of the lectures,
- R manual,
- assignments.

An introductory book on statistics (containing the prerequisite knowledge for this course) is for example
- Statistical reasoning for everyday life, J.O. Bennett, W. Briggs, M.F.Triola.

For more background on the topics in this course, the following books are recommended:
- Linear models with R, by J.J. Faraway (emphasis on the implementation in R);
- Extending the linear model with R, by J.J. Faraway (emphasis on the implementation in R);
- A first course in the design of experiments; a linear models approach, by D.C. Weber and J.H. Skillings (emphasis on the designs, also implementation in SAS).

Vereiste voorkennis
Introductory statistics.

Aanbevolen voorkennis
Probability and statistics courses.

Doelgroep
mAI, mCS

Overige informatie
All assignments are to be solved using the statistical package R (http://www.r-project.org/).

Green Lab

Vakcode	X_418158 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. I. Malavolta
Examinator	dr. I. Malavolta
Docent(en)	dr. I. Malavolta
Lesmethode(n)	Hoorcollege, Practicum
Niveau	400

Doel vak
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.

Inhoud vak
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.

Onderwijsvorm
Lectures (H). Lab sessions (pra).

Toetsvorm
Teamwork. Project assignments (pro).

Literatuur
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in software engineering. Springer. Material distributed on Blackboard.

Aanbevolen voorkennis
Basic statistical analysis techniques (descriptive statistics and most common tests).

Doelgroep
mCS, PDCS, mAI

High Performance Computing and Big Data

Vakcode	XMU_40013 ()
Periode	Periode 3
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Niveau	400

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2016-2017/zoek-vak/vak/1545337

History of digital cultures

Vakcode	XMU_418107 ()
Periode	Periode 3
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	O.W. Schrofer
Examinator	O.W. Schrofer
Niveau	400

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2016-2017/zoek-vak/vak/28352

Overige informatie
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.

ICT4D: Information and communication technology for Development

Vakcode	X_405101 ()
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. K.S. Schlobach
Examinator	dr. K.S. Schlobach
Docent(en)	dr. K.S. Schlobach, dr. V. de Boer
Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	400

Doel vak
In the developed world Computers are ubiquitous, and ICT has rapidly grown into a critical asset for economic, technological, scientific and societal progress. The main objectives of this course are:
1) to make the next generation of Computer Scientists aware of:
a) The importance of ICTs for the developing world and the unexpected way developing countries are leapfrogging into the information age
b) The opportunities and challenges that exist for an information scientist in the area of 'development4development'
c) The influence of context in a typical ICT4D project
d) The complexity of deploying an ICT project within a development context, and how to tackle this.
2) to equip the students with some initial project management, technological and programming skills specific to an ICT deployment in a developing country.
Positioned at the heart of the VU's vision of social relevance as one of the guiding principles, the core aim of the course is to raise the awareness that we as Computer Scientists can make a significant difference by sharing our expertise according to well established principles of international development.

Inhoud vak
This course gives an introduction to the relatively new field of ICT4D and will be given jointly by experts from the Department of Computer Science (CS) and the Center for International Cooperation (CIS) with lecturers from both backgrounds who will focus on their areas of expertise.

In the course we will give an overview over methodology, technology and the social dimension of the usage of Information Technology in the

context of Development. We will introduce a general framework for ICT4Development. Subsequently, lecturers from CIS will teach you how to analyse a development problem and introduce the analytical methods required for an indepth understanding of a potential development support project. Lecturers from CS will provide some initial technological knowledge required for running an ICT project in a developing country, such as Voice technology or database technology on small, inexpensive, hardware. It will give an overview over technology already applied, such as specific networks, connection types, hardware as well as specific software environments, but also introduce basic concepts in project management for ICT projects.

In lectures, you will first be introduced to a number of tools, techniques and programming languages that can be used for ICT4D projects. We will introduce case studies, highlight real-world ICT4D projects, both from inside and outside academia. We will discuss requirements and strategies used in the projects. We will present a number of initiatives in which the VU is involved in. To prepare for the lectures, you will read related literature provided by the lecturers.

In the tutorial lectures, students will first get familiar with the tools and techniques introduced in the practical lectures. We will assess your skills in assignments.

Onderwijsvorm
The course will be a combination of lectures and project work.

Toetsvorm
Practical assignment

Literatuur
Collection of papers.

Doelgroep
mAI, mCS, mIS

Individual Systems Practical

Vakcode	XM_405088 ()
Periode	Ac. Jaar (september)
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. S. Voulgaris
Examinator	dr. S. Voulgaris
Niveau	500

Doel vak
Give very motivated students the opportunity to work on challenging, research-oriented projects, with a strong focus on serious systems work.

Inhoud vak
The content is to be negotiated individually with the professor supervising and grading it. Agreeing with a professor at the VU is absolutely necessary *before* the project starts. Showing up in

retrospect, trying to earn some credits for work you did last summer or on a side-job, will certainly not qualify.

Only a few students (the ones that come up with a convincing project proposal) will be allowed to do it. Proposals that are not challenging enough, are not deep into systems work, or their nature prevents the clear assessment of the student's own contribution, will be declined.

Toetsvorm
Supervised systems work.

Doelgroep
mCS, mPDCS

Industrial Internship

Vakcode	XM_405080 ()
Periode	Ac. Jaar (september)
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Niveau	400

Doel vak
The student applies study contents in an industrial setting. After completion of the internship, the student will be able to reflect on the peculiarities of industrial practice, in relation to knowledge and skills obtained at university.

Inhoud vak
Individual project work by which the student applies the study contents in an industrial setting.

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. The project must aim at applying state-of-the-art methods and technology and must require an original contribution by the student.

Before the start of the internship, the student has to get approval for the internship project by a VU or UvA Computer Science lecturer. 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.

The report must contain both a description of the project and a reflection on study contents being applied in an industrial context. For the project, the problem, the student’s solution, and an evaluation of the results must be presented. For the reflection part, (at least) the following question need to be addressed: 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?

The overall grade is a weighted average of the project contents and the student's solution (50%), the quality of the written report (35%), and the quality of the reflection of study vs. industrial context (15%).

Internship assignments that either do not give the student sufficient freedom to devise his or her own original solution or that can not be reported upon due to disclosure limitations by the company are not eligible for this course.

Onderwijsvorm
individual project work in an industrial setting

Toetsvorm
written report

Aanbevolen voorkennis
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.

Doelgroep
mCS, mPDCS

Overige informatie
Various lecturers

Information Visualization

Vakcode	XMU_418143 ()
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2016-2017/zoek-vak/vak/23828

Overige informatie
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.

Internet programming

Vakcode	X_405082 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. S. Voulgaris
Examinator	dr. S. Voulgaris
Docent(en)	dr. S. Voulgaris

Lesmethode(n)	Hoorcollege
Niveau	400

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
Good knowledge of both C and Java

Doelgroep
mAI, mCS, mPDCS

Introduction to Computational Science

Vakcode	XMU_418111 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Niveau	400

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2015-2016/zoek-vak/vak/18229

Overige informatie
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.

Kernel Programming

Vakcode	XM_40014 ()

Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. K. Razavi
Examinator	dr. K. Razavi
Docent(en)	C. Giuffrida, dr. K. Razavi
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
Gaining a deep understanding of kernel programming and of major OS subsystems design. This course is very tough and time-consuming.

Inhoud vak
The course will feature a number of hands-on assignments accompanied by lectures on advanced operating system kernel design and programming concepts. In each assignment, students will be expected to start with a minimal kernel implementation and exercise their kernel hacking skills on one of the major operating subsystems (i.e., memory management, drivers, etc.). This will involve programming in both C and assembly as well as directly interfacing with the hardware. The course will also link lectures and assignments to modern operating system features and offer insights into state-of-the-art OS research efforts.

Onderwijsvorm
Hoorcollege and practical

Literatuur
Slides and online material

Knowledge and Media

Vakcode	X_405065 (405065)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. T. Kuhn MSc
Examinator	dr. T. Kuhn MSc
Docent(en)	dr. T. Kuhn MSc
Lesmethode(n)	Werkcollege
Niveau	500

Doel vak
The goal of the course is to provide insights in the concepts of information organization, knowledge representation, ontologies, and knowledge processes in relation to various ICT-based media.

Inhoud vak
This course treats the principles and theories that form the foundation of information organization and knowledge-intensive processes, and puts them in relation to various media applications. Knowledge processes are those processes that use knowledge
(reasoning), document knowledge (representation), acquire knowledge or transfer knowledge (teaching). The relation between knowledge processes and media will be explored, and various types of applications will be discussed.

Onderwijsvorm
Working lectures

Toetsvorm
Portfolio

Literatuur
Articles announced through Blackboard

Knowledge Engineering

Vakcode	X_405099 ()
Periode	Periode 2+3
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. A.C.M. ten Teije
Examinator	dr. A.C.M. ten Teije
Docent(en)	dr. A.C.M. ten Teije
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
goals:
1) to be able to elicitate knowledge from experts by using several elicitation techniques
2) to be able to build all CommonKads models that play a role in the development of a knowledge based system, this includes the context of the KBS and the expertise model based
3) to be able to implement the expertise model as a prototype
4) to be able to reflect on your own process of modelling and building a knowledge based system, and to reflect on your product (=which are the models and the implementation)

Inhoud vak
Knowledge Engineering is a discipline that involves integrating knowledge into a program for solving a complex problem, which requires human expertise. Typical tasks are classification, diagnosis, planning etc. In the course we use CommonKADS as the methodology for the process of modeling the organisation, the context and the knowledge intensive tasks.
This methodology give clear guidelines and concrete templates for modeling the organisational aspects and the expertise model, which is the core model of knowledge based system. The notion of pattern-based knowledge modeling is a key issue in the knowledge modelling process.

The goal of the final project is to perform the entire knowledge technology process for a knowledge intensive problem of your own choosing, starting with context analysis, up to a (partial) implementation of the knowledge based system.

Onderwijsvorm
Lectures, assignments, group project

Toetsvorm
Assignment, project reports.

Literatuur
Schreiber, Akkermans, Anjewierden, de Hoog, Shadbolt, van de Velde, Wielinga: Knowledge Engineering & Management. The MIT Press, Cambridge MA, 2000, ISBN 0-262-19300-0.

Doelgroep
mAI, mIS, mCS-TAI

Lambda Calculus

Vakcode	XMU_418108 ()
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2015-2016/zoek-vak/vak/21590

Overige informatie
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.

Large Scale Data Engineering

Vakcode	X_405116 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. P.A. Boncz
Examinator	prof. dr. P.A. Boncz
Docent(en)	prof. dr. P.A. Boncz
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
The goal of the course is to gain insight into and experience with algorithms and infrastructures for managing big data.

More information is found on http://event.cwi.nl/lsde

Inhoud vak
This course confronts the students with some data management tasks, where the challenge is that the mere size of this data causes naive solutions, and/or solutions that work only on a single machine, to stop being practical. Solving such tasks requires the computer scientist to have insight in the main factors that underlie algorithm performance (access pattern, hardware latency/bandwidth), as well as possess certain skills and experience in managing large-scale computing infrastructure.

Onderwijsvorm
There are two lectures per week, and requires significant practical work. The practicals are done outside lecture hours, at the discretion of the students who are supported remotely through Skype screen sharing.

Toetsvorm
In the first assignment the students can work either on their own laptops via a prepared VM, or in the cloud using an Amazon EC2 Micro Instance; and there is an online competition between practicum teams for the best result. The second assignment, using a Hadoop Cluster, are done on the SurfSARA Hadoop cluster (90 machines, 720 cores, 1.2PB storage). For this assignment, a report of 5-8 pages must be written. The students also need to read two scientific papers of choice, related to the second assignment, and present these in class. There is no written examn; the grade is based on the two assignments grades, the grade for the in-class presentation and attendance/participation.

Literatuur
scientific papers provided in the course

Vereiste voorkennis
Hadoop environments are consist of Linux machines, so some basic ability in working with these comes in handy. Also, you must have some programming skills in C,C++ or Java.

Aanbevolen voorkennis
Programming proficiency in C/C++ or Java

Doelgroep
mCS, mPDCS

Literature Study and Seminar

Vakcode	X_405111 ()
Periode	Ac. Jaar (september)
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. P. Lago
Examinator	prof. dr. P. Lago
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
After taking this course, students will be able to:
- design a sound desk research based on scientific literature and identify the relevant literature related to the target research; or perform an in-depth study of a selected set of literature sources;
- criticize, analyze, and discuss scientific literature;
- reflect on the in-depth knowledge gained during the course on the selected literature study topic.

Inhoud vak
The course consists of carrying out a literature study on a topic chosen in agreement with a selected tutor.

The ‘theoretical preparation’ of the course consists of studying the provided material on literature study design, where the student learns how to go from a research question to a well-structured analysis of the literature. This step can be carried out by translating the research question into a sound search query, identifying adequate on-line literature search engines, and performing a motivated selection of the literature for further analysis; or alternatively by selecting, together with the tutor, a set of relevant literature and performing an in-depth study.

The actual literature study starts with the 'exploration phase' in which the student must identify a topic of interest or a course particularly appreciated. He or she will then contact the person in charge of the identified research area/course and discuss with him/her the possibility to carry out a literature study under his/her supervision.

In general the literature study will be either predominantly broad, leading to a literature survey on the selected theme, or deep, carving out the intricacies of a specific topic. The outcome of the literature study is a final report, which must include: study design; overview of selected literature; analysis of the literature; discussion and conclusions. At the end of this phase, the student gives one final presentation to the research group of the tutor. This presentation includes research questions, study design, study execution, and discussion of analysis.

Onderwijsvorm
Individual assignment.

Toetsvorm
Final report (V). Final presentation (pre).

Grading criteria: quality of study design; rationale for literature selection (if applicable); quality of results (quality of writing, scientific quality of the analysis, discussion of the findings, reflection in the drawn conclusions, clear answer to main research question); quality of technical report (style, clarity, organization); correctness and completeness of references and citations; final presentation to the research group where he/she carries out the literature study.

Literatuur
Material on literature study design online at
https://wiki.cs.vu.nl/mp/index.php/LSresources

Doelgroep
mCS

Overige informatie
The student should look for a tutor in the area he or she would like to carry out the literature study. See online material (Course reading) for further information.

Lecturers:
Staff members of the CS Department.

Logical Verification

Vakcode	X_400115 (400115)
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. J.C. Blanchette
Examinator	dr. J.C. Blanchette
Docent(en)	dr. F. van Raamsdonk
Lesmethode(n)	Hoorcollege, Practicum
Niveau	500

Doel vak
Introduction to the proof assistant Coq and its type-theoretic foundations.

Inhoud vak
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.

Onderwijsvorm
2 times 2 hours theory class, 2 times 2 hours practical work

Toetsvorm
Written exam,
obligatory Coq-exercises,
obligatory hand-in theory exercises,
possibly presentations of papers.

Literatuur
Course notes possible with some recent papers as addition.

Vereiste voorkennis
An introduction course in logic.

Doelgroep
mCS, mAI, mMath, mPDCS

Master Project

Vakcode	XM_400442 (400442)
Periode	Ac. Jaar (september)
Credits	36.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Niveau	500

Doel vak
The Master Project is the place in the study where scientific and professional skills are trained most extensively. The Master Project will always involve an element of originality or creativity, for example in performing a design task or in contributing to the solution or the analysis of a scientific problem. Other important elements of the Master Project are the cooperation with professionals and possibly with other students, planning the project, and documenting and presenting the final results.

Inhoud vak
The Master project concludes the Master programme. It is either in the form of a graduation project in one of the research groups of the Department of Computer Science of VU University or the Informatics Institute of the University of Amsterdam, or as an internship in a company. In most cases, it will be performed as an individual project but it can be a group project as well.

Onderwijsvorm
The Master Project has always to be supervised by a staff member of the Department of Computer Science of VU University or the Informatics Institute of the University of Amsterdam. In the case of an internship, supervision will be in cooperation with a supervisor at the company. Internships proposed by the student him- or herself need advance approval from a member of staff, who will become the project supervisor.

Toetsvorm
The final grade will be based on the quality of the research, the written thesis and an oral presentation. In the case of a group project, students have to write individual theses and give individual oral presentations, demonstrating their individual proficiency.

Vereiste voorkennis
The student must have completed (almost) all other courses of the Master programme.

Doelgroep
mCS

Overige informatie
For additional information and procedural rules we refer to the website of Faculty of Sciences of VU University.
There, you will also find links to the web pages of the research groups of the Department of Computer Science (VU) and the Informatics Institute (UvA), with options for master projects.
For company internships, useful documentation can be found on the website of the Internship Office.

Parallel Programming Practical

Vakcode	X_400162 (400162)
Periode	Periode 2+3
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. ir. H.E. Bal
Examinator	prof. dr. ir. H.E. Bal
Docent(en)	prof. dr. ir. H.E. Bal
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
Obtain practical experience with parallel programming using different programming systems.

Inhoud vak
During this practical, several parallel programs have to be written, using different programming environments, including Java, MPI, and Chapel. 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 presents for each problem the implementation approach taken and discusses the outcomes of the experiments conducted.

Onderwijsvorm
Practical computer work; students work on their own; there is one (kickoff) meeting and supervision from PhD students.

Toetsvorm
Practical computer work, final report.

Vereiste voorkennis
Knowledge of parallel programming in Java/Ibis, MPI, and Chapel (as taught in the Parallel Programming course) is required, as well as practical experience with C and Java.

Doelgroep
Masters Computer Science, PDCS, AI, and Computational Science

Overige informatie
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

Parallel System Architectures

Vakcode	XMU_40015 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Docent(en)	dr. A.D. Pimentel
Lesmethode(n)	Hoorcollege, Computerpracticum
Niveau	400

Doel vak
To understand the principles and design of modern (multi-core) microprocessor systems and how such processors can be aggregated, using interconnection networks, into parallel/distributed computing systems. Here, insight will be provided in both high-performance, general-purpose computer systems and more constrained embedded computer systems.

Inhoud vak
This course builds on a basic knowledge of microprocessor architecture. It develops this with and emphasis on instruction-level concurrency in microprocessor design and concurrency in both memory systems and parallel/distributed computing systems. The topics that are covered include superscalar and VLIW processor architectures, instruction- and thread-level parallelism, memory hierarchy, distributed- and shared-memory parallel computers, interconnection networks and new architecture trends.

Onderwijsvorm
Lectures and computer lab sessions.

Toetsvorm
Written examination (60%) and practical assignments (40%).

Literatuur
John Hennessy and David Patterson,'Computer Architecture: a Quantitative Approach', Morgan Kaufmann, 3rd ed. or later.

Vereiste voorkennis
Students need to have programming skills in C/C++, and basic knowledge on computer organization.

Intekenprocedure

Registration is required via https://www.sis.uva.nl before the start of the semester.
Please visit the website of your programme through http://student.uva.nl and check the A-Z list 'Course and Exam Registration' for more information.
Please note that in 2016-17 this course will be given at the VU University Amsterdam.

Overige informatie
Lecturer:
Dr. A.D. Pimentel

Performance Engineering

Vakcode	XMU_40016 ()
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Niveau	500

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2016-2017/zoek-vak/vak/1544979

Performance of Networked Systems

Vakcode	X_405105 ()
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. R.D. van der Mei
Examinator	prof. dr. R.D. van der Mei
Docent(en)	dr. ing. T. Kielmann, prof. dr. R.D. van der Mei
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
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.

Inhoud vak
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?

Onderwijsvorm
Classroom lectures and practical homework assignments.

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

Literatuur
Textbook, supplemented with a reader on Stochastic Performance Modelling.

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

Vereiste voorkennis
The students should have basic knowledge of computer networks.

Doelgroep
mBA, mCS, mPDCS, mEct

Programming Concurrent Systems

Vakcode	X_418109 ()
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen

Inhoud vak
This course has expired

Programming Large-scale Parallel Systems

Vakcode	XM_40017 ()
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen

Coördinator	prof. dr. ir. H.E. Bal
Examinator	prof. dr. ir. H.E. Bal
Docent(en)	prof. dr. ir. H.E. Bal
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
You will
- be introduced to the domain of High Performance Computing (HPC)
- learn about design methods for parallel algorithms
- compare different parallel computer architectures
- analyze performance of network topologies
- develop basic familiarity with a range of parallel programming constructs,
environments and languages
- gain insight into some selected parallel applications

Inhoud vak
This lecture discusses how programs can be written that run in parallel on a large number of processors, with the main 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 MPI message passing) and more advanced techniques like parallel object-oriented approaches from the Java ecosystem or dedicated HPC programing languages (like Cray's high productivity language Chapel) will be discussed. Several parallel applications are
discussed, including nearest-neighbor stencil computations, N-body simulations and search algorithms.

Onderwijsvorm
Lectures (4 hours per week), given by prof.dr.ir. Henri Bal (VU) and Dr Clemens Grelck (UvA). There is a separate Parallel Programming Practical (6 ECTS) in P2 and P3 whose aim is to complement the contents of this course with practical skills and experience. That course makes heavy use of our state-of-the-art DAS research cluster.

Toetsvorm
Written exam

Literatuur
Papers will be made available on Blackboard

Doelgroep
mAI, mBIO, mCS, mPDCS, m Computational Science

Overige informatie
Lecturers:
prof.dr.ir. Henri Bal (VU)
Dr. Clemens Grelck (UvA)

Programming Multi-core and Many-core Systems

Vakcode	XMU_40018 ()
Periode	Periode 4

Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Niveau	400

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2016-2017/zoek-vak/vak/1544980

Protocol Validation

Vakcode	X_400117 (400117)
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. W.J. Fokkink
Examinator	prof. dr. W.J. Fokkink
Lesmethode(n)	Hoorcollege, Practicum, Werkcollege
Niveau	500

Doel vak
Learning to use formal techniques for specification and validation of communication protocols.

Inhoud vak
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.

Onderwijsvorm
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.

Toetsvorm
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.

Literatuur
Wan Fokkink, Modelling Distributed Systems, Springer 2007.

Aanbevolen voorkennis
Logica en Modelleren

Doelgroep
mAI, mCS, mPDCS

Overige informatie
The course is taught once every two years, the next opportunity will be during study year 2017-2018.

Secure Software

Vakcode	XM_40019 ()
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. K. Razavi
Examinator	dr. K. Razavi
Docent(en)	prof. dr. ir. H.J. Bos, dr. K. Razavi
Lesmethode(n)	Hoorcollege
Niveau	500

Doel vak
This course covers security for software and mainly from the exploitation side.
This is not a generic course about computer security, but rather a focused course on software exploitation.

Specifically, students will need to carry out demanding assignments for exploiting existing software systems of a wide range of operation. From the lowest part, which is an OS kernel to the highest part, which is a web application, the browser, and mobile applications that run on smart-phones and tablets. Once students are familiar with the mechanics of exploitation for each particular domain, they will have a short guide to proposed defenses that aim at protecting software. At the end of the course, students will have deep understanding of how systems work and how practical exploitation can be carried out in full at different software domains.

Inhoud vak
This is mainly a hands-on course for exploitation. The students are assumed to be fairly familiar with security concepts and basic exploitation techniques (for example how memory bugs can be used for attacking programs). During the course they will have the opportunity to combine basic security concepts for carrying out advanced exploitation of real-world software systems in full.

Specifically, the course focuses on advanced exploitation techniques that can compromise software systems of different domains. Students need to complete 4 self-contained assignments during the course. Three assignments are oriented towards advanced exploiting of applications in three different domains: (a) a Linux kernel, (b) a mobile device, (c) a web application and the browser. The fourth assignment targets understanding and implementing defenses that could potentially protect the systems that are compromised in assignments (a)-(c).

Onderwijsvorm
Lectures and (very challenging) practical assignments.

Toetsvorm
Written exam (30%) and practical assignments (70%).

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

Vereiste voorkennis
Knowledge of C is probably essential.

Aanbevolen voorkennis
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.

Doelgroep
mCS, mPDCS.

Serious Games

Vakcode	X_405097 ()
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. A.P.W. Eliens
Examinator	prof. dr. A.P.W. Eliens
Docent(en)	prof. dr. A.P.W. Eliens
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
Serious games are more and more considered to be an effective means to bring about awareness, acquire skills, change behavior, and influence social patterns. With elementary game development technology, the students will explore the potential of serious games in a social context, using casual game mechanics, and what recently has been identified as the dynamics of gamification.

Inhoud vak
The course will cover the following topics:
* an introduction to game design
* practical skills in game development
* game mechanics and scoring mechanisms
* elementary game and utility theory
* media & communication theory
* game interaction patterns
* practical applications of serious games
Students are required to work in teams of 2-4 people, with as a goal the actual development of a serious game, with social network support.

Onderwijsvorm
lectures and practicum

Toetsvorm
essay and practicum assignment(s)

Literatuur
online reference material(s)

Aanbevolen voorkennis
preferably, but not obligatory, project interactive multimedia and multimedia authoring

Doelgroep
Choice for master students CS, IS, and others, with an interest in multimedia and game development

Overige informatie
For information and registration, see: www.cs.vu.nl/~eliens/serious

Service Oriented Design

Vakcode	X_405061 (405061)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. P. Lago
Examinator	prof. dr. P. Lago
Docent(en)	prof. dr. P. Lago
Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	400

Doel vak
Learn advanced design techniques applicable to large service-oriented software systems. Be able to select among them and apply them for a specific system. Be able to reason about and assess the design decisions.

Inhoud vak
The lectures explain the concepts related to the Service Orientation software paradigm and Service Oriented Architecture (SOA).
The lectures provide the students with knowledge about how to identify the requirements for a service-oriented software system, how to map them on business services and transform them into complex networks of software services. Special emphasis is given to the design reasoning techniques for crucial decision making, service identification, SOA design and migration. Each year experts from academia and industry are invited to give guest lectures.

The students participate in small teams to piecemeal develop understanding of various service-oriented aspects, and work on an assigned SOA design project.

Onderwijsvorm
Lectures (H), group work (w), project (pro).

Toetsvorm
Written reports of the assignments. Teamwork. Presentations.

Literatuur
Material handed out by the lecturer and on Blackboard.

Vereiste voorkennis
Software modeling (knowledge of UML and SoaML preferred).

Aanbevolen voorkennis
Programming.

Doelgroep
mAI, mCS, mIS

Overige informatie
Registration for this course is compulsory four weeks prior to the start. Further information on this module will be made available on the Blackboard system http://bb.vu.nl.

Software Architecture

Vakcode	X_400170 (400170)
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. P. Lago
Examinator	prof. dr. P. Lago
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
Get acquainted with the field of software and information architecture. Understand the drivers behind architectural decisions. Be able to develop and reason about an architecture of a non-trivial software system.

Inhoud vak
Students work in groups to develop an architecture for a fictitious system. They have to develop different representations (called views) of the architecture. These different representations emphasize different concerns of people that have a stake in the system. Each group will also be asked to assess ("test") the architecture of another group for certain quality attributes.

Onderwijsvorm
Lectures (H). Group work with a number of assignments (pro). Presentations (pre).

Toetsvorm

Project work (pro). Written exam (T).

Literatuur
Len Bass et al, Software Architecture in Practice, 3rd Edition, 2012

Doelgroep
mCS, mIS

Intekenprocedure
Registration is compulsory at least 4 weeks before course starts.

Software Asset Management

Vakcode	X_400412 (400412)
Periode	Periode 1
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	prof. dr. C. Verhoef
Examinator	prof. dr. C. Verhoef
Docent(en)	prof. dr. C. Verhoef
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
Gain insight in IT costs, benefits, risks, and returns on information technology.

Inhoud vak
In this class we treat several techniques to aid in software cost estimation. We discuss how IT migrates from cost issue to strategic asset, and how to come to grips with this important production factor. We provide insight in how to support decision making on IT investment issues. Examples from the Financial and Insurance industry, and independent software vendors are discussed. With IT benchmarks we obtain insight in the risks of IT developmtn and the operational costs of IT. We introduce the notion of an IT portfolio, and how to perform quantitative analyses with it, to aid in justifying IT investments.

Onderwijsvorm
Seminar with presentations of staff and students.

Toetsvorm
Essay on selected topics from articles. You can do this individually or in a group of 3 persons max.

Literatuur
Articles and chapters from books.

Aanbevolen voorkennis
Proficiency in software engineering and statistics.

Doelgroep
mCS, mIS, mBA, mAI

Software Testing

Vakcode	X_400439 (400439)
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. N. Silvis-Cividjian
Examinator	dr. N. Silvis-Cividjian
Docent(en)	dr. N. Silvis-Cividjian
Lesmethode(n)	Hoorcollege, Practicum
Niveau	400

Doel vak
- Familiarization with basic terminology in software testing.
- Familiarization with techniques and tools used for test generation, execution and adequacy measument.
- Familiarization with software testing literature in a specific area by independent reading of selected research publications.

Inhoud vak
Testing is a method to improve software quality. Realistically, software testing is a trade-off between budget, time and quality. It is impossible to test everything so choices have to be made. Students learn how to make these choices and systematically test a software product based only on its requirements or when the code is also available.
This course provides an introduction to software testing with an emphasis on technical activities like test generation, selection, execution and assessment. The course tries to answer a few questions like: How to design test cases? When to automate testing? When to stop testing? What to
test when a new version of the product is ready? How to test a safety critical software? How to predict how many faults are in a program? During their practical assignments the students have to develop and test object-oriented software using the techniques learned in class.
A few guest lectures showing examples of testing in industry are also planned.

Topics: boundary value analysis, equivalence partitioning, combinatorial testing, model based
testing, control-flow testing, data-flow testing, mutation
testing, regression testing, inspections, automated testing.

Onderwijsvorm
Lectures and compulsory homework assignments and mini-project.

Toetsvorm
Compulsory practical assignments and written exam.

Literatuur
A. Mathur, Foundations of software testing, Addison-Wesley Professional; 2 edition (February 13, 2014), 2014, *ISBN: * 978-8131794760

Aanbevolen voorkennis
Programming skills in Java or Python

Doelgroep
mCS, mAI

Overige informatie
All material is available in Blackboard.

Software Testing Practical

Vakcode	X_405124 ()
Periode	Periode 6
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. N. Silvis-Cividjian
Examinator	dr. N. Silvis-Cividjian
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
After completing this practical, the students will be able to:
• Develop and test software for different embedded systems, and produce artifacts such as: requirements specifications, safety and risk analysis, design documents, behavioral models, code, test plans and test reports.
• Monitor and analyze their development and testing process in order to asses or compare different software testing techniques

Inhoud vak
During the Software Testing course in period 5, students have been exposed to a wide range of traditional and novel testing techniques. During their practical homework assignments, they could apply some of these techniques on small software artifacts. However, there was not enough opportunity to exercise all these techniques in a realistic, industrial-like setting. Moreover, since there is no one silver-bullet, good-for-all cases testing technique, the students never investigated which technique or approach is better for a certain type of software-under-test (SUT). Therefore, it seems very useful to create in period 6 a "playground" where students can experiment with different testing techniques they learned about during the Software Testing course.
The purpose of this practical is to involve students in a realistic software development and testing process, that starts from requirements specification and ends up with high quality code. The SUT benchmark consists of software for microcontroller-based systems, such as: model-train railways, ph control installations and autonomous robots.
The students will work in groups of 4 students. Each group will develop and
test one of these SUTs, while continuously gathering test progress information. By analyzing these data, they will end up by drawing conclusions on the strengths and weaknesses of different testing techniques. Examples of possible experiments are: test-driven-development (TDD) vs. waterfall development model, formal specifications

vs. natural language specifications, random testing vs. systematic testing, the strengths and weaknesses of model-based testing, combinatorial testing, static analysis, defect prediction, etc. Each group will write a report and will present their findings in class. Testing expertise from software industry will be available to steer the student groups, and thus bring the academic setup closer to a realistic, industrial setting.

Onderwijsvorm
Practical lab sessions, weekly meetings with a steering group

Toetsvorm
Written report and presentation

Vereiste voorkennis
A Software Testing course. Programming skills in Java, Matlab, C or Python. Familiarization with various testing tools.

Doelgroep
mCS

Overige informatie
All material is available in Blackboard. The number of participants is limited to 24.

Term Rewriting Systems

Vakcode	XM_400121 (400121)
Periode	Periode 4+5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	drs. J. Endrullis
Examinator	drs. J. Endrullis
Docent(en)	drs. J. Endrullis
Lesmethode(n)	Hoorcollege, Werkcollege
Niveau	400

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

Inhoud vak
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

Onderwijsvorm
Lectures and practice sessions

Toetsvorm
Written examination

Literatuur
Course notes will be provided

Doelgroep
mCS, mPDCS, mAI, mMath

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

The Social Web

Vakcode	X_405086 ()
Periode	Periode 4
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. D. Ceolin
Examinator	dr. D. Ceolin
Docent(en)	dr. D. Ceolin
Lesmethode(n)	Hoorcollege, Computerpracticum
Niveau	400

Doel vak
In this course the students will learn theory and methods concerning communication and interaction in a Web context. The focus is on distributed user data and devices in the context of the Social Web.

Inhoud vak
This course will cover theory, methods and techniques for:

- personalization for Web applications
- Web user & context modelling
- user-generated content and metadata
- multi-device interaction
- usage of social-web data

Onderwijsvorm
- lectures
- practical sessions
- assignments including final paper

Toetsvorm
Weighted average of group assignments and final individual paper

Literatuur

- course lecture slides
- selected articles, videos and Web links for each lecture

Aanbevolen voorkennis
Basic programming skills

Doelgroep
VU: mIS
UvA: master Information Studies - Human-Centered Multimedia
mCS
mAI

Watson Innovation

Vakcode	X_405129 ()
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	dr. L.M. Aroyo
Examinator	dr. L.M. Aroyo
Docent(en)	dr. L.M. Aroyo, A. Dumitrache MSc
Lesmethode(n)	Hoorcollege, Computerpracticum
Niveau	400

Doel vak
The Watson Innovation course is a collaboration between the Vrije Universiteit, University of Amsterdam and IBM Netherlands. It offers a unique opportunity to learn about IBM Watson, cognitive computing and the meaning of such artificial intelligence systems in a real world and big data context. Students from Computer Science and Economics faculties will join their complementary efforts and creativity in cross-disciplinary teams to explore the business and innovation potential of these technologies.

Inhoud vak
- Basics of Cognitive Computing and IBM Watson
- Understanding the original IBM Watson
- Develop ideas for Cognitive Computing apps
- Build real prototypes using IBM Watson technologies
- Showcase your ideas to real clients.

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

Toetsvorm
Evaluation of group projects and individual peer-reviews

Literatuur
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/)

Doelgroep
A balanced mix of Computer Science, AI, Information Science, Business Analytics and Business & Economics students (from VU as well as UvA) in their 3rd year of bachelor or master level.

Intekenprocedure
Places are limited, so sign up as soon as possible. For questions please contact b.timmermans@vu.nl or oana.inel@vu.nl

Overige informatie
Lecturer(s)
dr. L.M. Aroyo, B. Timmermans, O.Inel, A. Dumitrache

Web Data Processing Systems

Vakcode	XM_40020 ()
Periode	Periode 2
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Coördinator	J. Urbani
Examinator	J. Urbani
Docent(en)	J. Urbani
Lesmethode(n)	Hoorcollege
Niveau	400

Doel vak
After taking this course, you will be able to:

- Understand the fundamentals of the most important problems that modern Web companies face daily;
- Process large amounts of Web data efficiently using state-of-the-art tools that are currently used in the Web industry;
- Extract useful insights from raw data available on the Web;
- Adapt or reuse techniques used on the Web to other fields (e.g. Data Mining, Artificial Intelligence) where similar problems might occur.

Inhoud vak
The Web constitutes the largest repository of knowledge that is available to mankind, and its impact on modern society is unprecedented at many levels. Many Web companies are valued with billion dollar quotations and are now central to our modern life.

The key players in the Web industry must face numerous challenges that are concerned with the size, distribution, heterogeneity, and the uncontrolled nature of the Web. Systems to process Web data require the application of a combination of techniques spanning databases, distributed systems, data mining, and artificial intelligence.

The goal of this course is to introduce the student to the most advanced systems and techniques which deal with Web data. Important classes of problems concern:

- the storage and retrieval of Web data (How can we store and retrieve

information from large social networks, graphs, or large volumes of text?)
- efficient entity disambiguation (What is a particular web page talking about?)
- large-scale knowledge extraction (What sort of knowledge can we extract from web documents -- e.g. Wikipedia?)
- effective link prediction (Is there a connection between two users/events/concepts?)
- expressive ontological inference (Can current knowledge lead to more implicit knowledge?)
- trust (Can we trust the content on a certain blog post?)

This course will describe techniques to perform these tasks with a particular emphasis on scalability, which is a crucial aspect in this domain. In order to better understand the challenges and effectiveness of current solutions, the student will be called to implement practical assignments on realistic Web data. These assignments will be part of the final evaluation of the course.

Onderwijsvorm
The course takes the form of lectures and practical assignments.

Toetsvorm
A combination of exams and group homework assessments.

Literatuur
A mixture of scientific publications and other material available on the Web.

Doelgroep
XM_CS 1, XM_PDCS 1

Web Services and Cloud-based Systems

Vakcode	XMU_418110 ()
Periode	Periode 5
Credits	6.0
Voertaal	Engels
Faculteit	Faculteit der Exacte Wetenschappen
Niveau	400

Inhoud vak
http://studiegids.uva.nl/xmlpages/page/2015-2016/zoek-vak/vak/20645

Doelgroep
mCS

Overige informatie
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.