Program overview

06-Mar-2025 15:05

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Code	Omschrijving	ECTS	p1 p2 p3 p4 p5
Master CS 2021	**Master Computer Science 2021**		
Track Software Technology 2021			
Common Core ST 2021			
CS4015	Behaviour Change Support Systems	5	
CS4065	Multimedia Search and Recommendation	5	
CS4200-A	Compiler Construction	5	
CS4220	Machine Learning 1	5	
IN4150	Distributed Algorithms	6	
IN4152	3D Computer Graphics and Animation	5	
IN4191	Security and Cryptography	5	
IN4252	Web Science & Engineering	5	
IN4315	Software Architecture	5	
IN4343	Real-time Systems	5	
IN4344	Advanced Algorithms	5	
Specialistievakken start kwartaal 1 2021	**Specialisation courses start first period 2021**		
AP3421	Fundamentals of Quantum Information	4	
CS4070	Multivariate Data Analysis	5	
CS4200-A	Compiler Construction	5	
CS4215	Quantitative Performance Evaluation for Computing Systems	5	
CS4270	Conversational Agents	5	
EE4C06	Networking	5	
ET4388	Ad-hoc Networks	5	
IN4010(-12)	Artificial Intelligence Techniques	6	
IN4049TU	Introduction to High Performance Computing	6	
IN4191	Security and Cryptography	5	
IN4252	Web Science & Engineering	5	
IN4307	Medical Visualization	5	
IN4344	Advanced Algorithms	5	
IN4387	System Validation	5	
WM-ITAV-4010	Scientific Writing	2	
Specialistievakken start kwartaal 2 2021	**Specialisation courses start second period 2021**		
CS4015	Behaviour Change Support Systems	5	
CS4055	High Performance Data Networking	5	
CS4090	Quantum Communication and Cryptography	5	
CS4200-B	Compiler Construction B	5	
CS4220	Machine Learning 1	5	
IN4089	Data Visualization	5	
IN4150	Distributed Algorithms	6	
IN4302TU	Building Serious Games	5	
IN4341	Performance Analysis	5	
Specialistievakken start kwartaal 3 2021	**Specialisation courses start third period 2021**		
AP3132	Advanced Digital Image Processing	6	
CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5	
CS4135	Software Verification	5	
CS4160	Blockchain Engineering	5	
CS4195	Modeling and Data Analysis in Complex Networks	5	
CS4210-A	Algorithms for Intelligent Decision Making	5	
CS4225	Educational Technologies	5	
CS4230	Machine Learning 2	5	
CS4235	Socio-Cognitive Engineering	5	
CS4240	Deep Learning	5	
CS4275	Web Programming Languages	5	
CS4400	Deep Reinforcement Learning	5	
CS4405	Analysis of Concurrent and Distributed Programs	5	
CS4415	Sustainable Software Engineering	5	
CS4430	Network Security	5	
EE4560	Information Theory	5	
ET4394	Wireless IoT and Local Area Networks	5	
IN4152	3D Computer Graphics and Animation	5	
		5	

IN4253ET	"Hacking Lab"-Applied Security Analysis	
IN4315	Software Architecture	5
IN4325	Information Retrieval	5
IN4343	Real-time Systems	5
IN4391	Distributed Systems	5
Specialistievakken start kwartaal 4 2021	**Specialisation courses start fourth period 2021**	
CS4035	Cyber Data Analytics	5
CS4065	Multimedia Search and Recommendation	5
CS4125	Seminar Research Methodology for Data Science	5
CS4140ES	Embedded Systems Laboratory	5
CS4145	Crowd Computing	5
CS4205	Evolutionary Algorithms	5
CS4210-B	Intelligent Decision Making Project	5
CS4265	Computer and Network Security: Advanced Topics	5
CS4280	Language-Based Software Security	5
CS4290	Seminar on Distributed Machine Learning Systems	5
CS4295	Release Engineering for Machine Learning Applications	5
CS4410	Category Theory for Programmers	5
EE4715	Array Processing	5
ET4030	Error Correcting Codes	4
ET4285	Measuring and Simulating the Internet	4
IN4185	Globally Distributed Software Engineering	5
IN4254	Smart Phone Sensing	5
IN4255	Geometric Data Processing	5
IN4331	Web-scale Data Management	5
IN4333	Language Engineering Project	5
Track Data Science & Technology 2021		
Common Core DST 2021		
CS4035	Cyber Data Analytics	5
CS4065	Multimedia Search and Recommendation	5
CS4220	Machine Learning 1	5
IN4010(-12)	Artificial Intelligence Techniques	6
IN4089	Data Visualization	5
IN4252	Web Science & Engineering	5
IN4315	Software Architecture	5
IN4344	Advanced Algorithms	5
IN4391	Distributed Systems	5
Specialistievakken start kwartaal 1 2021	**Specialisation courses start first period 2021**	
AP3421	Fundamentals of Quantum Information	4
CS4070	Multivariate Data Analysis	5
CS4200-A	Compiler Construction	5
CS4215	Quantitative Performance Evaluation for Computing Systems	5
CS4270	Conversational Agents	5
EE4C06	Networking	5
ET4388	Ad-hoc Networks	5
IN4010(-12)	Artificial Intelligence Techniques	6
IN4049TU	Introduction to High Performance Computing	6
IN4191	Security and Cryptography	5
IN4252	Web Science & Engineering	5
IN4307	Medical Visualization	5
IN4344	Advanced Algorithms	5
IN4387	System Validation	5
WM-ITAV-4010	Scientific Writing	2
Specialistievakken start kwartaal 2 2021	**Specialisation courses start second period 2021**	
CS4015	Behaviour Change Support Systems	5
CS4055	High Performance Data Networking	5
CS4090	Quantum Communication and Cryptography	5
CS4200-B	Compiler Construction B	5
CS4220	Machine Learning 1	5
IN4089	Data Visualization	5
IN4150	Distributed Algorithms	6
IN4302TU	Building Serious Games	5
IN4341	Performance Analysis	5
Specialistievakken start kwartaal 3 2021	**Specialisation courses start third period 2021**	
AP3132	Advanced Digital Image Processing	6
CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
CS4135	Software Verification	5
CS4160	Blockchain Engineering	5
CS4195	Modeling and Data Analysis in Complex Networks	5
CS4210-A	Algorithms for Intelligent Decision Making	5
CS4225	Educational Technologies	5
		5

CS4230	Machine Learning 2	
CS4235	Socio-Cognitive Engineering	5
CS4240	Deep Learning	5
CS4275	Web Programming Languages	5
CS4400	Deep Reinforcement Learning	5
CS4405	Analysis of Concurrent and Distributed Programs	5
CS4415	Sustainable Software Engineering	5
CS4430	Network Security	5
EE4560	Information Theory	5
ET4394	Wireless IoT and Local Area Networks	5
IN4152	3D Computer Graphics and Animation	5
IN4253ET	"Hacking Lab"-Applied Security Analysis	5
IN4315	Software Architecture	5
IN4325	Information Retrieval	5
IN4343	Real-time Systems	5
IN4391	Distributed Systems	5
Specialistievakken start kwartaal 4 2021	**Specialisation courses start fourth period 2021**	
CS4035	Cyber Data Analytics	5
CS4065	Multimedia Search and Recommendation	5
CS4125	Seminar Research Methodology for Data Science	5
CS4140ES	Embedded Systems Laboratory	5
CS4145	Crowd Computing	5
CS4205	Evolutionary Algorithms	5
CS4210-B	Intelligent Decision Making Project	5
CS4265	Computer and Network Security: Advanced Topics	5
CS4280	Language-Based Software Security	5
CS4290	Seminar on Distributed Machine Learning Systems	5
CS4295	Release Engineering for Machine Learning Applications	5
CS4410	Category Theory for Programmers	5
EE4715	Array Processing	5
ET4030	Error Correcting Codes	4
ET4285	Measuring and Simulating the Internet	4
IN4185	Globally Distributed Software Engineering	5
IN4254	Smart Phone Sensing	5
IN4255	Geometric Data Processing	5
IN4331	Web-scale Data Management	5
IN4333	Language Engineering Project	5
Track Artificial Intelligence Technology 2021		
Common Core AIT 2021		
CS4065	Multimedia Search and Recommendation	5
CS4205	Evolutionary Algorithms	5
CS4210-A	Algorithms for Intelligent Decision Making	5
CS4220	Machine Learning 1	5
CS4240	Deep Learning	5
CS4270	Conversational Agents	5
IN4010(-12)	Artificial Intelligence Techniques	6
IN4315	Software Architecture	5
IN4325	Information Retrieval	5
Specialistievakken start kwartaal 1 2021	**Specialisation courses start first period 2021**	
AP3421	Fundamentals of Quantum Information	4
CS4070	Multivariate Data Analysis	5
CS4200-A	Compiler Construction	5
CS4215	Quantitative Performance Evaluation for Computing Systems	5
CS4270	Conversational Agents	5
EE4C06	Networking	5
ET4388	Ad-hoc Networks	5
IN4010(-12)	Artificial Intelligence Techniques	6
IN4049TU	Introduction to High Performance Computing	6
IN4191	Security and Cryptography	5
IN4252	Web Science & Engineering	5
IN4307	Medical Visualization	5
IN4344	Advanced Algorithms	5
IN4387	System Validation	5
WM-ITAV-4010	Scientific Writing	2
Specialistievakken start kwartaal 2 2021	**Specialisation courses start second period 2021**	
CS4015	Behaviour Change Support Systems	5
CS4055	High Performance Data Networking	5
CS4090	Quantum Communication and Cryptography	5
CS4200-B	Compiler Construction B	5
CS4220	Machine Learning 1	5
IN4089	Data Visualization	5
IN4150	Distributed Algorithms	6
IN4302TU	Building Serious Games	5

IN4341	Performance Analysis	5
Specialistievakken start kwartaal 3 2021	**Specialisation courses start third period 2021**	
AP3132	Advanced Digital Image Processing	6
CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
CS4135	Software Verification	5
CS4160	Blockchain Engineering	5
CS4195	Modeling and Data Analysis in Complex Networks	5
CS4210-A	Algorithms for Intelligent Decision Making	5
CS4225	Educational Technologies	5
CS4230	Machine Learning 2	5
CS4235	Socio-Cognitive Engineering	5
CS4240	Deep Learning	5
CS4275	Web Programming Languages	5
CS4400	Deep Reinforcement Learning	5
CS4405	Analysis of Concurrent and Distributed Programs	5
CS4415	Sustainable Software Engineering	5
CS4430	Network Security	5
EE4560	Information Theory	5
ET4394	Wireless IoT and Local Area Networks	5
IN4152	3D Computer Graphics and Animation	5
IN4253ET	"Hacking Lab"-Applied Security Analysis	5
IN4315	Software Architecture	5
IN4325	Information Retrieval	5
IN4343	Real-time Systems	5
IN4391	Distributed Systems	5
Specialistievakken start kwartaal 4 2021	**Specialisation courses start fourth period 2021**	
CS4035	Cyber Data Analytics	5
CS4065	Multimedia Search and Recommendation	5
CS4125	Seminar Research Methodology for Data Science	5
CS4140ES	Embedded Systems Laboratory	5
CS4145	Crowd Computing	5
CS4205	Evolutionary Algorithms	5
CS4210-B	Intelligent Decision Making Project	5
CS4265	Computer and Network Security: Advanced Topics	5
CS4280	Language-Based Software Security	5
CS4290	Seminar on Distributed Machine Learning Systems	5
CS4295	Release Engineering for Machine Learning Applications	5
CS4410	Category Theory for Programmers	5
EE4715	Array Processing	5
ET4030	Error Correcting Codes	4
ET4285	Measuring and Simulating the Internet	4
IN4185	Globally Distributed Software Engineering	5
IN4254	Smart Phone Sensing	5
IN4255	Geometric Data Processing	5
IN4331	Web-scale Data Management	5
IN4333	Language Engineering Project	5
Suggested Track Courses AIT 2021		
IN4089	Data Visualization	5
Seminar Courses CS & Literature Survey 2021		
CS4120	Seminar Science and Methods in Cyber security	5
CS4125	Seminar Research Methodology for Data Science	5
CS4130	Seminar Programming Languages	5
CS4165	Seminar Social Signal Processing	5
CS4210-B	Intelligent Decision Making Project	5
CS4245	Seminar Computer Vision by Deep Learning	5
CS4285	Seminar: Decentralized Systems	5
CS4290	Seminar on Distributed Machine Learning Systems	5
IN4306	Literature Survey	10
IN4310	Seminar Computer Graphics	5
IN4314	Seminar Selected Topics in Multimedia Computing	5
IN4326	Seminar Web Information Systems	5
IN4334	Analytics and Machine Learning for Software Engineering	5
IN4398	Advanced Practical IoT and Seminar	5
Free Elective Space 2021	**Suggested Free Elective Space 2021**	
Quantum Computing 2021		
AP3421	Fundamentals of Quantum Information	4
AP3432	Quantum Hardware 1 - Theoretical Concepts	4
AP3442	Quantum Hardware 2 - Experimental State of the Art	4
CS4090	Quantum Communication and Cryptography	5
EE4575	Quantum Computing Architecture and Electronics - Fundamentals and state-of-the-art	5
Language Courses & Skills 2021		
TPM018A	English Grammar for the University	2
		2

TPM303A	Intermediate Writing in English for the University	
TPM304A	Advanced Writing in English for the University	2
TPM305A	Writing a Masters Thesis in English	2
WM1115TU	Dutch Elementary 1	3
WM1116TU	Dutch Elementary 2	3
WM1117TU	Dutch Intermediate 1	3
WM1135TU	Advanced English for the University	3
Projects 2021		
TUD4040	Joint Interdisciplinary Project	15
Thesis Project		
IN5000	Final Project	45
Research Groups 2021		
Algorithmics 2021		
CS4205	Evolutionary Algorithms	5
CS4210-A	Algorithms for Intelligent Decision Making	5
CS4210-B	Intelligent Decision Making Project	5
CS4400	Deep Reinforcement Learning	5
IN4344	Advanced Algorithms	5
Computer Graphics and Visualisation 2021		
IN4089	Data Visualization	5
IN4152	3D Computer Graphics and Animation	5
IN4255	Geometric Data Processing	5
IN4302TU	Building Serious Games	5
IN4307	Medical Visualization	5
IN4310	Seminar Computer Graphics	5
Cyber Security 2021		
CS4035	Cyber Data Analytics	5
CS4090	Quantum Communication and Cryptography	5
CS4120	Seminar Science and Methods in Cyber security	5
CS4150	Systems Security	5
CS4185	Capstone Cyber Security	5
CS4265	Computer and Network Security: Advanced Topics	5
CS4430	Network Security	5
IN4191	Security and Cryptography	5
IN4253ET	"Hacking Lab"-Applied Security Analysis	5
UT-191612680	Computer Ethics	5
UT-192110940	Secure Data Management	5
UT-201100022	Cyber Crime Science	5
UT-201500038	E-Law	5
UT-201500039	Security Verification	5
UT-201500040	Introduction to Biometrics	5
UT-201500041	Cyber Security Management	5
UT-201500042	Privacy Enhancing Technologies	5
Cyber Security/SERG 2021		
CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
Distributed Systems 2021		
CS4160	Blockchain Engineering	5
CS4215	Quantitative Performance Evaluation for Computing Systems	5
CS4285	Seminar: Decentralized Systems	5
CS4290	Seminar on Distributed Machine Learning Systems	5
IN4150	Distributed Algorithms	6
IN4391	Distributed Systems	5
Embedded and Networked Systems 2021		
CS4055	High Performance Data Networking	5
CS4140ES	Embedded Systems Laboratory	5
CS4425	Visible Light Communication & Sensing	5
ET4285	Measuring and Simulating the Internet	4
ET4388	Ad-hoc Networks	5
ET4394	Wireless IoT and Local Area Networks	5
IN4254	Smart Phone Sensing	5
IN4343	Real-time Systems	5
IN4390	Quantitative Evaluation of Embedded Systems	5
IN4398	Advanced Practical IoT and Seminar	5
Interactive Intelligence 2021		
CS4015	Behaviour Change Support Systems	5
CS4125	Seminar Research Methodology for Data Science	5
CS4165	Seminar Social Signal Processing	5
CS4235	Socio-Cognitive Engineering	5
CS4270	Conversational Agents	5
IN4010(-12)	Artificial Intelligence Techniques	6
Multimedia Computing 2021		
CS4065	Multimedia Search and Recommendation	5
CS4195	Modeling and Data Analysis in Complex Networks	5
IN4314	Seminar Selected Topics in Multimedia Computing	5

Pattern Recognition & Bioinformatics 2021

CS4070	Multivariate Data Analysis	5
CS4176	Algorithms for network-based bioinformatics	5
CS4220	Machine Learning 1	5
CS4230	Machine Learning 2	5
CS4240	Deep Learning	5
CS4245	Seminar Computer Vision by Deep Learning	5
CS4250	Selected Topics in Molecular Biology	5
CS4255	Algorithms for sequence-based Bioinformatics	5
CS4260	Machine Learning in Bioinformatics	5
CS4329	Recent topics in bioinformatics	5

Programming Languages 2021

CS4130	Seminar Programming Languages	5
CS4135	Software Verification	5
CS4200-A	Compiler Construction	5
CS4200-B	Compiler Construction B	5
CS4275	Web Programming Languages	5
CS4280	Language-Based Software Security	5
CS4405	Analysis of Concurrent and Distributed Programs	5
CS4410	Category Theory for Programmers	5
IN4333	Language Engineering Project	5
IN4387	System Validation	5

QCE/ Network Architectures and Services 2021

EE4396	Mobile Networks	5
ET4034	Telecom Business Architectures and Models	4
IN4341	Performance Analysis	5

Software Engineering 2021

CS4295	Release Engineering for Machine Learning Applications	5
CS4405	Analysis of Concurrent and Distributed Programs	5
CS4415	Sustainable Software Engineering	5
IN4185	Globally Distributed Software Engineering	5
IN4315	Software Architecture	5
IN4334	Analytics and Machine Learning for Software Engineering	5

Web Information Systems 2021

CS4145	Crowd Computing	5
CS4225	Educational Technologies	5
IN4252	Web Science & Engineering	5
IN4325	Information Retrieval	5
IN4326	Seminar Web Information Systems	5
IN4331	Web-scale Data Management	5

Special Programmes 2021

Information Architecture 2021

IN4252	Web Science & Engineering	5
IN4325	Information Retrieval	5
IN4331	Web-scale Data Management	5
SEN1121	Complex Systems Engineering	5
SEN1141	Managing Multi-actor Decision-making	5
SEN1611	I&C Architecture Design	5
SEN1622	I&C Service Design	5

Bioinformatics 2021

CS4070	Multivariate Data Analysis	5
CS4176	Algorithms for network-based bioinformatics	5
CS4195	Modeling and Data Analysis in Complex Networks	5
CS4205	Evolutionary Algorithms	5
CS4220	Machine Learning 1	5
CS4230	Machine Learning 2	5
CS4240	Deep Learning	5
CS4245	Seminar Computer Vision by Deep Learning	5
CS4250	Selected Topics in Molecular Biology	5
CS4255	Algorithms for sequence-based Bioinformatics	5
CS4260	Machine Learning in Bioinformatics	5
CS4290	Seminar on Distributed Machine Learning Systems	5
CS4329	Recent topics in bioinformatics	5
EE4C06	Networking	5
IN4010(-12)	Artificial Intelligence Techniques	6
IN4049TU	Introduction to High Performance Computing	6
IN4089	Data Visualization	5
IN4150	Distributed Algorithms	6
IN4252	Web Science & Engineering	5
IN4306	Literature Survey	10
IN4307	Medical Visualization	5
IN4315	Software Architecture	5
IN4325	Information Retrieval	5
IN4331	Web-scale Data Management	5
IN4344	Advanced Algorithms	5

IN4391	Distributed Systems	5
IN5000	Final Project	45
NB4130TU	Biologic	3

Cyber Security 2021

AP3421	Fundamentals of Quantum Information	4
CS4035	Cyber Data Analytics	5
CS4090	Quantum Communication and Cryptography	5
CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
CS4120	Seminar Science and Methods in Cyber security	5
CS4150	Systems Security	5
CS4160	Blockchain Engineering	5
CS4185	Capstone Cyber Security	5
CS4265	Computer and Network Security: Advanced Topics	5
CS4280	Language-Based Software Security	5
CS4430	Network Security	5
IN4191	Security and Cryptography	5
IN4253ET	"Hacking Lab"-Applied Security Analysis	5
IN5000	Final Project	45
TPM020A	Economics of Cybersecurity	5
TPM025A	User-Centred Security	5
TPM027A	Cyber Risk Management	5
TPM030A	Introduction to Cloud as Infrastructure: The effects of the new business of computing on practice	5
UT-191612680	Computer Ethics	5
UT-192110940	Secure Data Management	5
UT-201100022	Cyber Crime Science	5
UT-201500038	E-Law	5
UT-201500039	Security Verification	5
UT-201500040	Introduction to Biometrics	5
UT-201500041	Cyber Security Management	5
UT-201500042	Privacy Enhancing Technologies	5
UT-202000026	Secure Cloud Computing	5

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Master CS 2021	
Contact for students	MASTER COORDINATORS DATA SCIENCE & TECHNOLOGY Dr. M. Aniche + 31 (0)15 27 84570 M.FinavaroAniche@tudelft.nl Dr. K.A. Hildebrandt + 31 (0)15 27 81445 K.A.Hildebrandt@tudelft.nl MASTER COORDINATORS SOFTWARE TECHNOLOGY Dr. J.S. Rellermeyer J.S.Rellermeyer@tudelft.nl Building 28 Room E3.360 Dr.ir. R. Bidarra + 31 (0)15 27 84564 R.Bidarra@tudelft.nl MASTER COORDINATORS ARTIFICIAL INTELLIGENCE TECHNOLOGY Dr. J.C. van Gemert J.C.vanGemert@tudelft.nl Dr. C. Lofi C.Lofi@tudelft.nl
Introduction 1	The Computer Science programme consists of three tracks and three optional Special Programmes. Students may opt for a special program in Bioinformatics, Cyber Security or Information Architecture. It is a two-year programme (120 EC), and it is taught in English. The CS programme has three tracks: Data Science and Technology (DST), Software Technology (ST) and Artificial Intelligence Technology (AIT). Students in the Software Technology track focus more on designing and engineering software artefacts, while students in the Data Science and Technology track focus more on answering research questions using sophisticated data analysis techniques. NOTE TU Delfts involvement in the EIT programme was discontinued as of academic year 2019/2020.

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Track Software Technology 2021

Introduction 1

In the Software Technology track of the Master of Computer Science program, the engineering of complex software systems takes on a central role. In this track, you will acquire knowledge and skills to design, develop and implement efficient algorithms, large-scale data structures and complex architectures. Furthermore, you will learn how to integrate them in real-world information-processing systems. Illustrative topics include distributed, multimedia, knowledge and secure processing, web and software engineering, visualization and interaction.

Software technology has a major impact on the economies of industrialised countries. Information-processing systems provide the backbone for almost all administrative and logistic operations within commercial production, business and public administration. Without software, high-tech systems are useless, whether large (e.g., a self-driving car or a robot for computer-assisted surgery) or small (e.g., a smartphone or an ordinary electronic card payment system). Moreover, software permeates all aspects of our life in society, ranging from Internet-based services like e-mail, online games, social networks and cloud computing, to large-scale scientific computing systems, traffic control systems and wireless sensor systems. All these systems need to have effective algorithms for their correct operation, good performance, high reliability and a well-thought-out architecture to make them easy to build and to maintain. Such essential features very well describe the specific focus of the Software Technology track.

THE CURRICULUM
1. An Individual Exam Programma (IEP) in this track consists of
a. a common core,
b. courses offered by the faculty EEMCS,
c. a seminar offered by the programme CS or a Literature survey (IN4306)
d. free electives,
e. a thesis project (IN5000 Final project) worth 45 credits and
f. if required, homologation.

The IEP must be drawn up in agreement with the thesis coordinator of the research group in which the student wishes to carry out his or her thesis project. The thesis coordinator is a member of the scientific staff of that research group.

the seminar of the research group in which the thesis is performed or the Literature Study (IN4306) is part of said IEP,

Free elective courses
c. the number of credits spent on free electives in said IEP is no higher than 25 credits,
d. the number of credits spent on homologation in said IEP is no higher than 15 credits,
e. at least 40 credits of the courses in the IEP (notwithstanding the thesis project) should be computer Science courses. A list of these courses is published annually in the digital study guide.

Free electives - language course list:
Up to 3 credits may be spent on language courses. These may only be chosen if required. Placement tests showing the necessity to take one or more of these courses must be taken and submitted to the master coordinator.

WM1101TU English for academic purposes-3 3
WM1135TU English for academic purposes-4 3
WM1136TU Written English for technologists-1 3
WM1102TU Written English fortechnologists-2 3
WM1137TU Spoken English for technologists-1 2
WM1112TU Spoken English for technologists-2 2
WM1115TU Elementary 1 Dutch for foreigners 3
WM1116TU Elementary 2 Dutch for foreigners 3
WM11117TU Dutch intermediate 1 3

The free elective space may also be used for an extra project:
TUD4040 Joint Interdisciplinary Project (JIP) 15

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Common Core ST 2021	
Introduction 1	Common Core MSc CS - ST (at least 25 EC): Choose 5 out of 11. NOTE: IN4315 Software Architecture 2018-2019 - Started your masters programme in 2017-2018 or before? The general rule is that students follow the masters programme of the year in which they have started their programme. So for example students who started their masters programme in 2017-2018 follow the Teaching and Examination Regulations (TER) of 2017-2018. In 2017-2018 course IN4315 Software Architecture was only part of the common core list of DST. ST students still may choose this course as a specialisation course. - Started your masters programme in 2018-2019? Students who started their masters programme in 2018-2019 follow the Teaching and Examination Regulations (TER) of 2018-2019. From 2018-2019 both DST and ST students may choose IN4315 Software Architecture as a common core course.

CS4015	Behaviour Change Support Systems	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	none	
Course Contents	Behavior change support systems (BCSS) are computer-based systems that support individuals to form, alter or reinforce cognitions, attitudes or behaviors without using coercion or deception. They can serve individuals throughout the various stages of a change process, such as awareness developing, contemplation, action strategy development, development of new behaviors, and maintaining these new behaviors. Virtual healthcare coaches, negotiation support systems, and applications that provide individuals with personalize financial guidance are three examples of these systems. To establish, modify or maintain change BCSS can deploy computerized persuasive strategies (e.g. reducing effort to establish target behavior, or argumentation and reflection strategies), simulations (e.g. serious gaming, virtual reality), relational software agents (e.g. ePartners, virtual coaches), and personalization based on longitudinal user data. BCSS are found in many domains, including education, sales, negotiation, management, and particular in the health domain.	
Study Goals	The course allows students to achieve understanding of principles, concepts and theories underlying BCCS systems and methods for designing them.	
Education Method	In the pre-recorded video material, theories, principles and methods are presented, discussed and illustrated with examples from the field. The video material is support by online self-tests. In the lectures, teacher and students discuss and make assignment related to pre-recorded material of that week. At home students work on their own in small groups on coursework assignments to develop a product design for a BCSS. In the practicum session student groups presented the progress on their coursework and receive feedback. Expected Workload Pre-recorded video material: 18 hours (2 hours × 9 lectures) lecture: 10 hours (10 × 1 hours) practicum 14 hours (7 × 2 hours) Reading time: 36 hours (9 × 4 hours reading time) Preparation presentation: 3 hours (3 × 1 hours for each presentation) Coursework project, including writing report, and final presentation: 50 hours Exam preparation and revision: 9 hours Total = 140 hours	
Literature and Study Materials	Will be announced on brightspace	
Books	Wendel, S. (2013). Designing for Behavior Change: Applying Psychology and Behavioral Economics. " O'Reilly Media, Inc.".	
Assessment	The course is assessed by coursework and an exam as follows: (60%) computerised examination (or oral exam) (40%) Coursework Project (resulting in a report, and final presentation include question and answer round where individual group members are assessed on coursework) If the expected number of students registering for exam or resit is small, the teacher might decide to replace the computerized examination by an oral examination. Separate marks will be given for exam and coursework, only a combined mark is recorded in Osiris. A passing final grade for the course can only be earned when for both the exam and coursework at least a 5.0 is earned, and the weighted grade for exam and coursework is at least a 5.8. Resit coursework A second submission of modified coursework is only allowed for coursework that received a fail mark (<5.8) for the first submission. Overall resit mark for coursework will be capped to 5.8. Note that individual marks for coursework or computerised exam (or oral exam) do not carry to the next year. disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	2	
Co-Instructor	M.L. Tielman	

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4200-A	Compiler Construction	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	6/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	- programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-A covers the following topics: * Syntax and parsing - concrete syntax, abstract syntax - context-free grammars - derivations, ambiguity, disambiguation, associativity, priority - parsing, parse trees, abstract syntax trees, terms - pretty-printing - parser generation - syntactic editor services * Static semantics and type checking - name binding, name resolution, scope graphs - types, type checking, type inference, subtyping - unification, constraints - semantic editor services	
Study Goals	After this course, students should be able to: * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define the syntax of a programming language using declarative syntax definition that describes the concrete and abstract syntax of a programming language * Define basic editor services * Define the type system (name binding and typing rules) of a programming language using constraint generation rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the front-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%) - course project (50%)	

	Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	No materials are permitted during the exam.
Judgement	to be decided

CS4220	**Machine Learning 1**	**5**
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

IN4150	**Distributed Algorithms**	**6**
Responsible Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Knowledge of Computer Networks (CSE1405) and Operating Systems (CSE2430) is useful as background for understanding this course.	
Course Contents	Introduction to distributed algorithms; notions of time and ordering of events; distributed algorithms for message ordering, detecting global states, termination detection, deadlock detection, mutual exclusion, election, minimum-weight spanning trees, fault tolerance, consensus, and agreement; blockchain technology and its relation with consensus.	
Study Goals	Understand the main fundamental problems in distributed systems Understand the most important distributed algorithms that solve these problems Be able to reason about the execution of distributed algorithms Be able to program distributed algorithms Be able to select and summarize relevant literature on distributed algorithms	
Education Method	Lectures and lab work executed in groups of two students	
Literature and Study Materials	Lecture notes and lecture slides (available on Brightspace)	
Assessment	One paper summary and an in-person, on-campus written exam (closed book), which together are an integrated whole. The grade of this exam, which includes the grade of the paper summary (with a weight of 25%), is the grade for the course. In case in-person, on-campus written exams are not possible because of COVID-19, the written exam will be replaced by individual online oral exams.	
Permitted Materials during Tests	None except the list of algorithms	
Remarks	Lab work is 40 hrs.	

IN4152	3D Computer Graphics and Animation	5
Responsible Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. R. Marroquim	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students that haven't followed any previous Computer Graphics courses (like TI1806) will be able to participate, but might have to invest some more time to catch up in the first lectures.	
Course Contents	Have you ever wondered how Toy Story was made, why the game Last of Us 2 looks so beautiful, or have you ever wanted to create your own graphics application or game? Then you should consider following this course. If not, then you should still follow it... maybe, you will become interested! In this course, you will get a good idea of Computer Graphics in general. The topic is of very high relevance for the industry and the research community and has numerous applications in different domains, such as scientific visualization, video games, simulators, special effects, animated movies and many more. Here, you will learn about basic algorithms, as well as modern techniques. We will address several topics: the principles of image synthesis, object representations, geometric and hierarchical transformations, graphics cards and the graphics pipeline, realistic rendering (including global illumination and effects, such as reflections), expressive rendering, physics simulations, rendering control (including previsualization systems used by professionals in the movie industry), and perceptual rendering, which relies on properties of the human visual system to enhance the quality of the images. Besides course sessions on the theory of Computer Graphics, some of the algorithms will also be reproduced in practice, and deepened during the final project.	
Study Goals	The course teaches computer graphics techniques on an advanced level. After the course the student is able to classify the different modeling, shading, and display techniques. The student can reproduce the basic mathematical and algorithmic notions associated with these concepts, can comment on the weak and strong points of these techniques, and can apply the core concepts within a graphics program in practice.	
Education Method	lectures, instructions, research papers, lab work	
Literature and Study Materials	Research Papers in domain of selected topics, lecture sheets, online sources, optional books (see below)	
Books	Fundamentals of Computer Graphics by Shirley et al. - CRC Press Real-time Rendering by Tomas Akenine-Möller, Eric Haines, Naty Hoffman - Peters, Wellesley Real-Time Shadows by Elmar Eisemann, Michael Schwarz, Ulf Assarsson, Michael Wimmer - Taylor & Francis Computer Graphics. Principles and Practice by James D. Foley, Andries VanDam, Steven K. Feiner - Addison Wesley	
Assessment	The course will be evaluated with two grades, a project grade, accounting for 60%, and a paper grade 40%. The project grade is the result of a project and its presentation that is building upon the assignments that are handed out (roughly) weekly during the duration of this course. The paper grade is the result of the presentation of a scientific paper and the development of an associated practical implementation. Details of both elements will be presented during the lecture. Both grades (project and paper) have to be at least a 5.0 and their weighted average should be 6.0 or higher after rounding (0.5 steps).	

IN4191	Security and Cryptography	5
Responsible Instructor	Z. Erkin	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	UT-201500042 Privacy Enhancing Technologies (Q4)	
Expected prior knowledge	Basic understanding on the following is suggested. -Probability and statistics -Programming skills	
Course Contents	Motivation: Computers are now found in every layer of society, and information is being communicated and processed automatically on a large scale. Examples include medical and financial files, automatic banking, video-phones, pay-tv, teleshopping and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy. Synopsis: Security and cryptography are essential components of any digital system. In this course, the fundamentals of secure data storage and transportation of information are described. In particular, classical (e.g. Caesar, Vigenere) and modern encryption schemes (RSA, DES, AES, Elliptic curves) are described along with their mathematical background such as number theory. Methods for authentication, data integrity and digital signatures are discussed in detail, as these are the main components of many security architectures. The course also investigates more advanced topics such as zero-knowledge proofs and secret sharing schemes. Aim: It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security and privacy, as well as is familiar with present applications. Learning outcomes: The goal is to make students familiar with the basic concepts applied cryptography, including classical cryptography and modern secret key and public key cryptography. In particular, the students will acquire A sound understanding of the notion of security An understanding of the confidentiality, integrity and authenticity needs of the society Understand the role of cryptographic primitives including the differences between symmetric and asymmetric cryptography, the role of hash functions, digital signatures and PKI Understand the advanced topics in cryptography needed for the modern society with untrustworthy entities Among others things, the following topics are covered: -Classical systems -Information theoretic security -Definition of Security notions -Symmetric encryption (e.g. DES, AES) -Asymmetric encryption (RSA, Elliptic Curves) -Hash functions -Random number generation -Key Management -Digital Signatures, -*Secret Sharing. (if time permits) -*Zero Knowledge proofs (if time permits)	
Study Goals	It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security, as well as is familiar with present applications. Through assignments, students are expected to have the chance to work on the topics covered in the lectures.	
Education Method	Lectures, assignments and weekly exercises. Attention: This course requires full effort of 140 hours. Even more, if you lack the background (probability and modular arithmetic) Planned Workload: Lectures: 28 x 45minutes sessions, total 22 hours Practice session: 7 x 90 min. total 12 hours Assignments: 3 x 20 hour, total 60 hours Weekly study: 7 x 4 hours, total 28 hours Exam preparation: 20 hours Exam: 3 hours	
Literature and Study Materials	Cryptography made simple, Nigel P. Smart, 2nd Edition, Springer, 2016 (PDF Available Online) Handouts of lectures	
Assessment	Written exam(70%) + mandatory assignments (30%). Passing grade for the written exam and assignments is required. Please refer to CS regulations for further details. There is NO reparation for assignments. Points from previous years cannot be transferred. NOTE: the study guide information may change depending on the developments around the COVID-19 pandemic	
Exam Hours	In case of in person examination at campus: The exam is closed book. A cheat sheet of size A4, hand written is allowed for the written exam. Name and student number has to be present on each side. In case of remote exam from home: open book: textbook, slides and self-made notes only. randomised/customised exam.	

Permitted Materials during Tests	Only non-scientific calculators.

IN4252	**Web Science & Engineering**	**5**
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4343	Real-time Systems	5
Responsible Instructor	Dr. G. Iosifidis	
Instructor	Prof.dr. K.G. Langendoen	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures & 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	3TU MSc Embedded Systems; the corresponding courses are 2IN26 at TU Eindhoven, and 312030 at TU Twente	
Expected prior knowledge	Basic software engineering, C system programming, basic Linux operating system knowledge	
Course Contents	- basic concepts of RTS - worst case execution time estimation - scheduling policies - response-time analysis - jitter analysis - handling overload - multiprocessor scheduling - reservation-based scheduling	
Study Goals	The course intends to bring the student into the position to: - Explain the fundamental concepts and terminology of real-time systems - Construct task schedules using different scheduling policies under a given set of realistic system constraints - Analyze the timing behavior of a system for a given system model and scheduling policy - Discuss advantages and disadvantages of different scheduling policies for a given platform or system - Discuss the effect of hardware and software interferences on the timing behavior of a given system - Identify (reverse engineer) parameters of a scheduling scheme or a task set from output traces of the system - Derive (reverse engineer) the system specification from a given implementation (in the lab) - Evaluate the scheduling overheads of a given implementation (in the lab) - Implement event-based scheduling policies on a given microcontroller (in the lab)	
Education Method	lectures with exercises (32 hrs); self study (78 hrs); lab assignments (30 hrs)	
Books	Hard Real-Time Computing Systems by G.C. Buttazzo, Springer 2011	
Assessment	Written exam (grade) + lab work; the exam has a resit disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	3	
Permitted Materials during Tests	Simple calculator	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 1 2021

AP3421	Fundamentals of Quantum Information	4
Responsible Instructor	Dr. L. di DiCarlo	
Instructor	Dr. D. Elkouss Coronas	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Knowledge of linear algebra, probability and statistics.	
Course Contents	Approximate syllabus: - quantum states, unitary operations, and measurements; - universal gate sets; - entanglement, Bell test; - basic quantum communication protocols; - basic algorithms and quantum algorithmic techniques; - basic quantum error correction; - simple physical implementations of qubits.	
Study Goals	Motivation: Quantum information is the future of computing and communication. Quantum computers offer exponential speedup over any classical computer. Similarly, quantum communication offers many advantages, including the ability to create secure encryption keys where security rests only on the laws of nature. Synopsis: This class will teach you the fundamental principles of quantum information. You will learn essential concepts that distinguish quantum from classical devices. You will learn about quantum bits and the quantum operations and measurements that can be performed on them. You will learn the basic techniques used in quantum algorithms, and examine basic examples of such algorithms. You will also take the first step in understanding how a quantum bit can be physically implemented. Aim: To learn the fundamental concepts underlying quantum computation and communication systems.	
Education Method	3 hours of lecture, 1 hour tutorial per week.	
Literature and Study Materials	The main reference textbook for the course is Nielsen and Chuang, Quantum Computation and Information, Cambridge University Press.	
Assessment	30% homework assignments, 10% in class quiz, 60% final exam. A minimum grade of 5.0 (unrounded) for the final exam is required to pass the course.	
Permitted Materials during Tests		
Continuing Courses	This course forms part of the curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information AP3421-PR Quantum Information Project CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation	

CS4070	Multivariate Data Analysis	5
Responsible Instructor	Dr.ir. F.H. van der Meulen	
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	4/4/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	

Expected prior knowledge

* Introduction Probability Theory and Statistics: see for instance

A Modern Introduction to Probability and Statistics
Understanding Why and How
Series: Springer Texts in Statistics
Dekking, F.M., Kraaikamp, C., Lopuhaä, H.P., Meester, L.E.
2005, XVI, 488 p. 120 illus., Hardcover
ISBN: 1-85233-896-2

* Basic calculus

* Linear Algebra: matrix multiplication, the inverse of a matrix, the transpose of a matrix, least square solution.
see:

David C. Lay: Linear Algebra and Its Applications
ISBN-10: 0321385179 ISBN-13: 9780321385178
©2012 Pearson)

Course Contents

PART I:
Stochastic models will be developed on the basis of probability theory. Probability theory describes the behavior of certain phenomena in terms of how likely it is that certain values will occur. Central features of the models will be discussed are random variables, probability density functions, and the expected value operator. In describing random processes and signals, the correlation function and conditional probabilities play a central role.

It addresses the following subjects:
1. Random variables. Matlab exercise on estimation of PDF, expected value and variance.
2. Refresher correlation. Calculating with correlation functions.
3. Random processes, correlation function, stationarity, wide sense stationarity, estimation of correlation function (Matlab exercise).
4. Random signal processing, power spectral density function, white noise.
5. AR processes, linear prediction: theory and Matlab exercise.
6. Markov chains.

PART II:
A course in advanced statistics about linear models, Bayesian inference, classification problems, Gaussian processes and Markov Chain Monte Carlo.

Study Goals

PART I:
1. Probability Theory
- Conditional) probabilities, the law of total probability, and Bayes rule.
- Solve probability problems that require the use of axioms of probability.

2. Definition and Description of Random Variables and Processes
PDF, PMF, CDF, Covariance, Correlation- Determine if a given PDF, PMF, CDF, variance, (auto/cross-)correlation(-function), (auto/cross-)covariance(-function), power spectral density complies with (theoretical and analytical) requirements.
- Convert the description of a probabilistic problem into a probabilistic model using PDF, PMF, or CDF.

3. PDF/PMF and Expected Value
Calculate the various forms of expected value of (combinations of) random variables and random processes
- For a given (amplitude continuous/discrete and time continuous/discrete) probability model calculate the following probabilistic (marginal, joint and conditional) characterizations: PDF, PMF, CDF, probability of an event, expected value, variance, covariance, correlation, correlation coefficient, auto/crosscorrelation function, auto/crosscovariance function, (cross) power spectral density.
- Calculate the PDF, PMF, expected value and variance of a derived random variable.

4. Properties of Random Processes
- Independence, orthogonality, uncorrelated, whiteness, IID- Determine if random variables/processes have the following properties: independent, orthogonal, uncorrelated, white, Poisson, Gaussian, Bernoulli, Markov, IID, stationary, WSS, ergodic.
- Calculate the expected value, variance, auto/crosscorrelation(function), auto/crosscovariance(function), power spectral density of a linear combination of random variables and of a linearly filtered (WSS, amplitude discrete/continuous, time discrete/continuous) random process.

5. Large NumbersCentral limit theorem, law of large numbers
- Solve problems that require the use of the central limit theorem in an engineering context
- Explain the law of the large numbers in an engineering context.

6. Statistical Estimators
- Estimated mean, variance, and correlation function
- Given a set of outcomes, sample functions or realizations, calculate estimators for expected value, variance, and (auto-)correlation function.

12. Application to Engineering Problems and Simulations
- Select and translate a simple electrical engineering or computer science problem into mathematical probability model. The emphasis is on problems in signal and image processing, telecommunication, and media and knowledge technology. The class of probability models encompasses the following random variables/processes: Bernoulli, exponential, binomial, Poisson, Gaussian, uniform.
- Justify and reflect on the approach taken in calculating or simulating (MatLab) the following probabilistic properties: PDF, PMF, expected value, variance, autocorrelation function, autocovariance function.

	PART II: After finishing this course, the student is able to apply and derive statistical methods from both the frequentist and Bayesian perspective for - linear models - classification problems - clustering problems - Gaussian process regression The student is able to give a clear presentation about the underlying statistical theory. The student is able to compute several statistical characteristics by hand.
Education Method	PART I: Lectures, working groups (problem solving), laboratory work (a Matlab exercise) Workload is around 15 hours for attending lectures, 5 hours of reading study material and preparing lectures, 15 hours for the lab course, 20 hours for preparing the exam, 3 hours for the exam, and 8 hours for a final report (66 hours in total). PART II: Classes and weekly exercises.
Books	PART I: R.D. Yates and D.J. Goodman, "Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers", ISBN 0-471-17837-3, John Wiley and Sons, New York, 2005, Second Edition. PART II: Simon Rogers and Mark Girolami "A first coruse in machine learning, 2nd edition" Chapman & Hall From this book chapters 1--4, 8 and 9 will be covered.
Assessment	The final grade is the average of the grades you get for part (I) and (II). For part (I) there is a lab and written exam, where the grade is determined by the exam, and the lab assignment should be Passed. If you fail the lab assignment, you'll get a second chance to submit around the time the resit. For part (II), there will be an on-campus written exam. To pass the course, the average should be 5.8 or higher, and the grade of each individual part should be a 5.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.
Exam Hours	PART I: Online exam of 3 hours. PART 2: On campus 3 hour written exam PART II: Written exam of 3 hours.
Permitted Materials during Tests	PART I: Self made notes on a two-sided written A4 sheet. Calculator. PART II: none
Remarks	PART II: This course is particularly interesting for students that are interested in statistical exploratory and quantitative techniques to analyse multivariate data.

CS4200-A	Compiler Construction	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	6/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	- programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-A covers the following topics: * Syntax and parsing - concrete syntax, abstract syntax - context-free grammars - derivations, ambiguity, disambiguation, associativity, priority - parsing, parse trees, abstract syntax trees, terms - pretty-printing - parser generation - syntactic editor services * Static semantics and type checking - name binding, name resolution, scope graphs - types, type checking, type inference, subtyping - unification, constraints - semantic editor services	
Study Goals	After this course, students should be able to: * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define the syntax of a programming language using declarative syntax definition that describes the concrete and abstract syntax of a programming language * Define basic editor services * Define the type system (name binding and typing rules) of a programming language using constraint generation rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the front-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%) - course project (50%)	

	Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	No materials are permitted during the exam.
Judgement	to be decided

CS4215	Quantitative Performance Evaluation for Computing Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	Todays computing systems become ever complex, due to the rapid development of hardware and software technology. It is challenging to design and run computing systems that guarantee users performance requirements in a resource efficient way. Various quantitative methods are applied to capture such complex system dynamics and predict metrics of interests, from the designing phase of the systems to the runtime performance, e.g., job response times and system anomaly. To optimize the performance of computing systems, a deep understanding on those methods and their applications on the system design are essential. Having practical hand-on experience on designing experiments, deriving models, and validating results with benchmark systems will prepare students to tackle challenges of real world computing systems.	
Study Goals	LO1. Design full/fractional factorial experiments for multi-variate regression analysis, e.g., finding critical parameters for deep learning clusters LO2. Apply queueing theory to analyse and predict run-time performance of applications, e.g., the average response times of on-line ML training service LO3. Apply machine learning models to analyse and predict the system dependability, e.g,root cause analysis for machine failure. LO4. Conduct experiments to profile applications and extract their workload parameters on real systems, e.g., e.g., deep learning clusters LO5. Develop resource management policies and validate them on real computing systems, e.g., deep learning clusters	
Education Method	Lectures: 7 weeks X 2-4h Practical: Derive, validate and evaluate performance models and resource management strategies for a chosen system via homework and group project. Multiple types of computing and network systems can be chosen from. Deliverables include a report and group presentation.	
Books	1) Performance Modeling and Design of Computer Systems: Queuing Theory in Action by Mor. Harchol-Balter 2) Design and Analysis of Experiments by Douglas C. Montgomery, 3) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics.	
Assessment	Homework (40%): 2 individual homework Group project (60%): group project report and presentation disclaimer: information may change depending on the developments around the coronavirus.	

CS4270	Conversational Agents	5
Responsible Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Instructor	Dr. M. Bruijnes	
Instructor	Dr. P.K. Murukannaiah	
Instructor	M.L. Tielman	
Co-responsible for assignments	F. Broz	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic programming skills (e.g. Python and Java) Probability theory and statistics	
Course Contents	Chatbots, embodied and conversational virtual agents, and social robots are becoming more and more popular. Many people are owning an Alexa, Cortana or Echo or are talking to their virtual assistant on their phone. Indeed, such technologies have the potential of making our lives easier and relieve people from the more repetitive tasks. For example, it is imaginable that such systems are being used for financial applications by helping customers with frequently asked questions but also to advise them on in the long term more impactful decisions such as their pension plans. Further applications can be imagined in the area of healthcare and education, some of which are already in existence today. In this course, attention will be given to different verbal and nonverbal behavioral characteristics, like speech, intonation, gaze and gestures that humans show when communicating with both other people and machines. This behavior is then related to different dialogue functions, including turn-taking, addressing others, and backchanneling, that give shape to the communication process. This course introduces conversational agent technology. We cover agent related technologies which can be grouped into: Dialog Management NLP speech synthesis social robotics	
Study Goals	After this course you have learned to: 1) Apply relevant linguistic and psychological theory to conversational agent systems 2) Analyse human-human conversational data to better design ML models 3) Explain which components are part of a dialog system and what distinguishes rule-based from statistical dialog systems 4) Describe the design and implementation of state-of-the-art conversational agents, give examples of their application areas and analyse and discuss the limitations of current systems 5) Evaluate the effects of affect and embodiment on human-agent interaction 6) Create and evaluate a socially-aware conversational agent by applying state of the art tools and libraries	
Education Method	There are 2 lectures and 1 practicum scheduled per week. Students work in groups of 3-4 on a group project. Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours)	
Literature and Study Materials	We use the book "The conversational interface " by Michael McTear, Zoraida Callejas, David Griol. This book is freely available through the TU Delft library. https://link-springer-com.tudelft.idm.oclc.org/book/10.1007%2F978-3-319-32967-3 Other relevant material will be provided on Brightspace.	
Assessment	Online Examination (30%) Group Assignment (50%) (This assignment will result in a group report and a group online demonstration with individual question/answer part) Group presentation (20%) The exam and the assignments are both intermediate results, and will not count separately for the next academic year. Only the final grade will remain. A passing final grade for this course can only be earned when for the online examination and the group assignment at least a 5,0 is earned, and the average grade for both is at least a 5,8. Projectwork with a mark lower than 5.8 can be modified and resubmitted. The mark for resubmitted coursework will be capped to 5.8 Note that individual marks for projectwork or written exam do not carry to the next year. We further grade the labs as pass/fail. By a successful pass of all labs a bonus of 0.3 will be awarded towards the group assignment grade. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr.ir. W.P. Brinkman	

EE4C06	Networking	5

Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem
Contact Hours / Week x/x/x/x	4/0/0/0
Education Period	1
Start Education	1
Exam Period	1 2
Course Language	English
Course Contents	PART 1: Basics, concepts and computations of networks 1. Basics of networking & introduction to Network Science - what is a network? Representation of a graph, basics of graph theory, overview of the relatively new theory of complex networks, called Network Science. - important characterizers of a network (network metrics) - basic network/graph models - examples of real-world networks (airline transportation, the web and Internet, social networks, brain networks, etc.) and applications of network science 2. Concepts of networking - routing - Quality of Service (QoS) - traffic management and scheduling - network robustness (failure, cascading effects,...) - overlay networking and new aspects of networking such as interdependent networks PART 2: Applications and examples of networks (as listed below) will be taught (some of those by a guest lecturer). Ranging from year to year, a selection among the following will be covered: 1. Electrical networks (smart grids) 2. Networks on Chip (NoC) 3. Optical networks 4. Computer Networks (the Internet) 5. Mobile communication networks 6. Sensor networks 7. Biological networks 8. Social networks
Study Goals	The course on Networking aims to provide a general and basic introduction to the art of networking, that tries to unravel the operation and behavior of networks, both man-made (infrastructures such as the Internet and power grids) as well as networks appearing in nature (such as the human brain, biological networks and social human interactions). The course on Networking will introduce concepts of the new Network Science, that basically studies the interplay between, on the one hand, the processes (also called functions or services) on the network and on the other hand, the underlying topology, that is mostly changing over time as an evolving organism, rather than as given or fixed object. Network Science combines many disciplines such as graph and network theory, probability theory, physical processes, control theory and algorithms. After this course, students are expected to represent/abstract real-world infrastructural network (e.g. a communication system) as a complex network, understand the basic methods to analyze properties of networks and dynamic processes on networks. Students will also understand why processes on networks and design of networks are so complex. Finally, students may appreciate the fascinatingly rich structure and behavior of networks and may realize that much in the theory of networks still lies open to be discovered.
Education Method	Lectures, slides & homework
Assessment	written examination disclaimer: information may change depending on the developments around the coronavirus.

ET4388	Ad-hoc Networks	5
Responsible Instructor	Dr. R.R. Venkatesha Prasad	
Contact Hours / Week x/x/x/x	3/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Wireless communications and networking Computer communication principles, Layering principle of Computer Networks. Digital communication.	

Course Contents

IMPORTANT NOTICE

Please note that the prevailing conditions of the COVID19 may force us to modify the study guide information. It may change depending on the developments around the coronavirus. We may have to make changes to the teaching methodology, assessment, practical work, assignment and group activities. This will be instructed by the faculty/university management based on the orders of the government from time to time to protect the faculty and students.
The above applies to all the fields in this coursebase for this course.

The course will be offered ONLINE. Face to Face meeting may be possible depending on the situation [Safety of everyone is the highest priority].

Ad-hoc networks are formed in situations where mobile computing devices require networking applications when a fixed network infrastructure is not available or not preferred to be used. In such cases, mobile devices may possibly set up an ad hoc network themselves. Ad-hoc networks are decentralized, self-organizing networks and are capable of forming a communication network without relying on any fixed infrastructure.

Ad-hoc networks form a relatively new field of research. In this lecture, besides general introduction to ad-hoc networks and their applications, we will focus on state-of-the-art methods and technologies for forming an ad-hoc network and maintaining its stability despite the dynamics of the network.

The contents of the course are as follows:

Positioning and applications (Chapter 1, 2 & 3 of the textbook, these topics are basics & pre-requisites; And Chapter 5)
o Definition of ad-hoc networks
o Comparison with infrastructure based systems
o Typical applications
o Advantages and challenges
o Radio technologies for ad-hoc networks
o Wi-Fi, Zigbee, Bluetooth

Modelling ad-hoc networks
o Propagation models
o Topology models based on graph theory
o Degree and hopcount
o Connectivity theorems

MAC protocols for ad-hoc networks (Chapter 6, 10 of the textbook)
o Introduction to MAC protocols
o Issues and design goals
o Classification
o Directional, muli-channel MAC protocols
o Energy efficiency in MAC protocols
o Quality of service

Self organisation and Routing (Chapter 7, 8, 11 of the textbook)
o Flooding
o Node discovery, neighbour discovery
o Route establishment
o Topology maintenance, localisation
o Proactive, reactive and hybrid routing
o Typical protocols
o Energy efficiency in routing
o Broadcast and multicast
o Effects of mobility on connectivity and capacity
o Effect of nodes joining and leaving the network

Advanced issues in ad hoc networks
o Wireless sensor networks (Chapter 12 of the textbook and papers)
o Cooperation (Reference papers)
o Simulating ad hoc networks as part of project (optional: ns3, OMNET, OPNET)
o Energy Harvesting

Project presentations by students

Study Goals

By the end of this course students should be able to:
- Model the ad-hoc networks using Graphs.
- Describe the working principles of medium access control protocols for ad-hoc networks
- Explain the working principles, advantages and disadvantages of different classes of routing protocols for ad-hoc networks
- Choose various components to form a coherent ad hoc networking architecture
- Develop a simulator to evaluate the MAC and routing protocols for ad hoc networks
- Assess the suitability of ad-hoc networks for different communication needs and scenarios

Education Method

The course will be taught in lecture form. The presence of students at all lectures is required for optimum result. Students are required to participate actively in various forms of activities and peer-learning. New forms of teaching aids are used.

Literature and Study Materials

1. Textbook: Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004.

	2. Lecture notes consisting of slides presented at the lectures (Slides are only teaching aid and they are not substitute for textbooks, research papers, etc). 3. Some recent journal papers 4. Optional Reference Books 4.1. Distributed Algorithms, Nancy A. Lynch, Morgan Kaufmann, 1996 (for networking algorithms) 4.2. Ad Hoc Mobile Wireless Networks, Principles, Protocols and Applications by Subir Kumar Sarkar , C Puttamadappa , and T. G Basavaraju, Auerbach Publications, 2008. This book is avaliable online in the library. 4.3. Wireless Ad Hoc and Sensor Networks, A Cross-Layer Design Perspective by Jurdak, Raja, Springer, 2007. This book is avaliable online in the library. 4.4. Ad-hoc Networks: Fundamental Properties and Network Topologies, by Ramin Hekmat, Springer. 5. OPNET/ns-2 web pages, tutorials and video lectures
Books	Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004. However, I also use other materials from Internet and other books listed above.
Assessment	1. There will be written tests/examinations for this course. 2. The students will carry out a project in a group and submit a short report. 3. Participation in off-track discussions on Facebook/Brightspace/FeedbackFruits and wikis. Final score is based on marks obtained during tests, project, assignment (in groups) and bonus marks. All the details will be given in the first class. Breakup: 2 Tests + Final Exam = 55% Project 40% Self assessment + Reflection 3% Activities on Feedback Fruits or any Online platform 2% ====================== Changes due to COVID19 in 2021 ================ There are three chapters Chapter 1: Network modelling Chapter 2: MAC protocol Chapter 3: Routing + Lab: Simple experiments using laptops (individual) + simulations (group-wise) Marks Breakup -------------- Homework/Assignments/Group works Part A1: Group-wise assignment. 3 times (1 per chapter) -- 15 marks Part A2: Group-wise Q&A. 3 times (1 per chapter) -- 30 marks Part B1: Individual experiment+report Part B2: Group-wise simulations + report + demo Part B1 + Part B2 - 50 Marks Part C: Self-assessment + peer activities -- 5 marks Resit: Part A1 & A2 will not be repeated. Only Part B1 & B2 are allowed for the Resit this year, because of COVID19; and the projects should be done individually in Resit. === (More information will be given in the first class) disclaimer: Information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	Different conditions for different test/exams. Conditions will be informed 1 week before the exams/test.

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

IN4049TU	Introduction to High Performance Computing	6
Responsible Instructor	Prof.dr.ir. H.X. Lin	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear algebra (matrix and vector operations), numerical analysis (solution of a system of linear equations;some experience with a programming language (e.g., C) is preferred but not required.	
Course Contents	This course is intended for students who are interested in computing-intensive research. In the course, a number of algorithms that are being used within a diversity of research areas is considered. The scaling behaviour of these algorithms in case of an increasing problem size and/or an increasing number of processors, is analysed. Attention is paid to those aspects of computer architectures that are important to understand the resulting performance, such as the memory hierarchy and the interconnection network. By analysing a number of case studies (applications) with respect to their computing-intensive character, possible bottlenecks will be determined. Based on performance analysis, it will be indicated how the effect of those bottlenecks can be reduced. The goal is to learn how to get a high performance with the available hard/architecture. The lab exercises will be done on a cluster of computers, the DAS-5 system at TU Delft. The emphasis will be on designing efficient parallel algorithms and on the necessary optimalisation of the performance. During the lab exercises, the following types of problems will be elaborated on: a parallel Poisson solver, a parallel finite element simulation and a parallel N-body simulation. More information, such as handouts and slides, can be found the Brightspace.	
Course Contents Continuation	High Performance Computing, parallel programming, parallel algorithm	
Study Goals	1. Knowledge about high performance computer systems including parallel and distributed architectures, and programming models; 2. Basic knowledge about the concepts of data decomposition and parallel algorithms; 3. Knowledge about various high performance (numerical) methods and their parallelization; 4. Capable to implement parallel programs (using MPI) on cluster of computers and GPU (using Cuda); 5. Obtain some experience on performance analysis of parallel programs.	
Education Method	Lectures, computer lab exercise using MPI. As an option, answers to the bi-weekly quizzes can be handed in, and a maximum of one bonus point to the exam grade can be obtained.	
Computer Use	Lab exercises (mandatory): implementing (small) parallel programs with C, MPI and Cuda.	
Literature and Study Materials	Will be made available throughout the course and can be downloaded from the Brightspace.	
Assessment	Written exam (50%) + Lab work (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Via Osiris	

IN4191	Security and Cryptography	5
Responsible Instructor	Z. Erkin	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	UT-201500042 Privacy Enhancing Technologies (Q4)	
Expected prior knowledge	Basic understanding on the following is suggested. -Probability and statistics -Programming skills	
Course Contents	Motivation: Computers are now found in every layer of society, and information is being communicated and processed automatically on a large scale. Examples include medical and financial files, automatic banking, video-phones, pay-tv, teleshopping and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy. Synopsis: Security and cryptography are essential components of any digital system. In this course, the fundamentals of secure data storage and transportation of information are described. In particular, classical (e.g. Caesar, Vigenere) and modern encryption schemes (RSA, DES, AES, Elliptic curves) are described along with their mathematical background such as number theory. Methods for authentication, data integrity and digital signatures are discussed in detail, as these are the main components of many security architectures. The course also investigates more advanced topics such as zero-knowledge proofs and secret sharing schemes. Aim: It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security and privacy, as well as is familiar with present applications. Learning outcomes: The goal is to make students familiar with the basic concepts applied cryptography, including classical cryptography and modern secret key and public key cryptography. In particular, the students will acquire A sound understanding of the notion of security An understanding of the confidentiality, integrity and authenticity needs of the society Understand the role of cryptographic primitives including the differences between symmetric and asymmetric cryptography, the role of hash functions, digital signatures and PKI Understand the advanced topics in cryptography needed for the modern society with untrustworthy entities Among others things, the following topics are covered: -Classical systems -Information theoretic security -Definition of Security notions -Symmetric encryption (e.g. DES, AES) -Asymmetric encryption (RSA, Elliptic Curves) -Hash functions -Random number generation -Key Management -Digital Signatures, -*Secret Sharing. (if time permits) -*Zero Knowledge proofs (if time permits)	
Study Goals	It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security, as well as is familiar with present applications. Through assignments, students are expected to have the chance to work on the topics covered in the lectures.	
Education Method	Lectures, assignments and weekly exercises. Attention: This course requires full effort of 140 hours. Even more, if you lack the background (probability and modular arithmetic) Planned Workload: Lectures: 28 x 45minutes sessions, total 22 hours Practice session: 7 x 90 min. total 12 hours Assignments: 3 x 20 hour, total 60 hours Weekly study: 7 x 4 hours, total 28 hours Exam preparation: 20 hours Exam: 3 hours	
Literature and Study Materials	Cryptography made simple, Nigel P. Smart, 2nd Edition, Springer, 2016 (PDF Available Online) Handouts of lectures	
Assessment	Written exam(70%) + mandatory assignments (30%). Passing grade for the written exam and assignments is required. Please refer to CS regulations for further details. There is NO reparation for assignments. Points from previous years cannot be transferred. NOTE: the study guide information may change depending on the developments around the COVID-19 pandemic	
Exam Hours	In case of in person examination at campus: The exam is closed book. A cheat sheet of size A4, hand written is allowed for the written exam. Name and student number has to be present on each side. In case of remote exam from home: open book: textbook, slides and self-made notes only. randomised/customised exam.	

Permitted Materials during Tests	Only non-scientific calculators.

IN4252	**Web Science & Engineering**	**5**
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4307	Medical Visualization	5
Responsible Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0 lectures & 0/4/0/0 lab.	
Education Period	1 2	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic knowledge of linear algebra, calculus and programming is needed. This course (IN4307) has been designed to complement the courses Advanced Image Processing (ET4283) and Medical Imaging (AP2231TUD). However, these two courses are NOT pre-requisites.	
Course Contents	Theory and practice (Notice project extends to Q2) of medical visualization. This includes the following aspects: data acquisition basics, clinical practice; image processing, e.g., filtering, segmentation and measuremen; medical volume visualization; illustrative visualization; advanced visualization for complex modalities; interaction techniques for medical data; advanced applications.	
Study Goals	By the end of the course, you should be able to LO1: Explain medical visualization algorithms and their applicability to medical problems. LO2: Discuss the advantages and disadvantage of medical visualization algorithms. LO3: Build a medical visualization system for a given problem: a. Discuss a suitable visualization for a given medical problem. b. Implement the most suitable solution. c. Judge the performance of the implemented solution.	
Education Method	The course will be based on a combination of lectures and practical assignments. A final project will be developed in Q2	
Literature and Study Materials	Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications Bernhard Preim and Charl P. Botha (not mandatory)	
Assessment	The evaluation will be based on - a written (or oral if the number of students allows) exam (40%) - assignments during the semester (10%) - a final project (50%) The final project will be done during the 2nd quarter. Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The assignments will consist of small programming exercises and open questions, as preparation for the final project. The practical sessions will provide time for working on the assignments with assistance. The deliverables will be program code and/or answers to questions. The final project will be the design and implementation of a visualization system for a given medical problem. The final project will be carried out in teams. The deliverables for the final project will be a report (paper), the results (e.g., code) and a short video presenting the project (i.e. screencast). The written exam will be arranged at the end of the first quarter. You are allowed to have the slides and material of the course during the exam. No computer or laptop is allowed. The exam has a resit. The project will have a resit if the exam (NOT the resit exam) has been passed with a mark of 7.5 or higher and it will be on an individual basis. The project resit is not automatic and must be initiated by you within two weeks of the grades being published. Resit of a project will mean starting a new project. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	Notes and written material. No computers.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

IN4387	System Validation	5
Responsible Instructor	C.B. Poulsen	
Contact Hours / Week x/x/x/x	4,0,0,0 Lectures & 2,0,0,0 lab	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	Embedded Systems Masters	
Expected prior knowledge	There are no strict entry conditions for this course. However, prior knowledge of requirements analysis is recommended. Furthermore, a good basic knowledge about logic and set theory is extremely beneficial.	
Parts	Behavioural specification of sequential and parallel using labelled transition systems, process algebra, and abstract data types; model checking of such systems using the modal mu-calculus. Model-based testing.	
Summary	Everyone who ever designed an embedded system or a communication protocol involving several components executing simultaneously has experienced that such software is inherently susceptible to bugs. Typical problems that occur are race conditions, deadlocks, and unexpected interplay between different components. Due to the parallel nature of these systems, it is notoriously hard to detect such bugs using testing (for example, timing plays a crucial role). The following quote from the famous Dutch computer scientist Edsger W. Dijkstra illustrates a further problem with testing. Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence. Edsger W. Dijkstra In this course, we study model checking, which in contrast to testing can also be used to show the absence of bugs. Model checking is a technique in which we consider all states in (a model of) the system based on an abstract model. Based on this state space we verify whether the model satisfies the desired properties. Properties are typically derived from the requirements of the system. We will restrict ourselves to verification techniques that do not reason about timing (merely about the order in which event happen). Finally, we see how model-based testing can be used to show that an implementation conforms to the specification of the system.	
Course Contents	Behavioral Specification using Process Theory (Labelled Transition Systems, various notions of behavioral equivalence) and process algebra. Model checking the modal mu-calculus, and model-based testing using IOCO.	
Study Goals	Upon completion of the course: 1. The student knows the fundamental theory necessary for specifying the behavior of embedded systems and for reasoning about this behavior. 2. The student can describe simple systems using this theory. 3. The student can formally specify requirements and prove (or disprove) them on the behavior. 4. The student is able to model a concrete embedded system, and verify that it satisfies its requirements. 5. The student is able to show that an implementation of a system conforms to its specification.	
Education Method	Lectures + Programming Assignments + Practical Project The course is structured into two parts: 1. There will be weekly mandatory programming assignments in the first four weeks of course will be a small set of mandatory. The programming assignments are assessed as pass/fail. The programming assignments are due after the first four weeks of the course. 2. In the last four weeks of the course, you will self-organize into groups of (about) 4 students, and will develop and verify a model of an embedded system. You will write a report that documents your model and its development. There will be a written exam with programming assignments at the end of the course.	
Computer Use	The theory introduced in this lecture is at the heart of the mCRL2 tool set. This tool set can be used to specify and verify systems, and visualize them. To be able to carry out the project it is required that the mCRL2 tool set is installed on your laptop (or one of the TU Delft systems, if you do not have a laptop you can use). It is open source software, and is free of charge. The software can be obtained from https://www.mcrl2.org.	
Literature and Study Materials	The course is based on the book by Groote and Mousavi (see "Books"). All other materials will be published on Brightspace.	
Books	J.F. Groote and M.R. Mousavi. Modeling and Analysis of Communicating Systems. MIT Press, 2014. ISBN: 9780262027717 (Chapters 1-7,11 are mandatory)	
Assessment	The result of this course will be based upon the results of the written examination (50%) and the practical project (50%). For both the programming exam and the practical project, a minimum of 5.0 is required in order to pass the course. To be eligible for taking the exam you must submit and pass the mandatory programming assignments for the first four weeks of the course. Grades of the project or written exam do not automatically carry over from previous years, so upon retaking the course talk to your lecturer first. For the exam a resit is scheduled. Please note that the study guide information for this course may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	The exam will be a 3 hour written exam with programming questions. You are allowed to use the book and any other static resources. You are not allowed to communicate or discuss exam questions with anyone but members of the teaching team for the course. Discussing or copying code will be considered fraud, and is reason for expulsion from the course.	
Enrolment / Application	Brightspace	
Co-Instructor	E. Visser	

WM-ITAV-4010	Scientific Writing	2
Module Manager	L. Meester	
Instructor	A. Glasbergen-Plas	
Instructor	M. Looij	
Instructor	M.J.Y. Wackers	
Instructor	S. Baars	
Instructor	M. Bliekendaal	
Instructor	L.C. Schroten	
Instructor	Drs. W.J. Blokzijl	
Instructor	Drs. B.M.D. van der Laaken	
Instructor	Drs. P.C. Post	
Instructor	Drs. A.E. Kam	
Co-responsible for assignments	Drs. B.M.D. van der Laaken	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1 2	
Exam Period	none	
Course Language	English	
Course Contents	In this course, you learn to write a scientific article, either a research article based on your own research data or a literature review about a subject of your own choice. This is a necessary skill for anyone who wants to pursue an academic career after their graduation, but it can also be used immediately for all academic texts you will write during your Master programme, such as your Master thesis. In seven weeks, we will go through all steps of the writing process, from formulating a good main question and finding relevant literature, to the actual writing, re-writing and final editing. You will exercise with finding, reading and managing relevant academic literature, writing in an academic style, building a comprehensive argumentation, reviewing fellow students' articles and using other students and the instructor's comments to improve your own work.	
Study Goals	The purpose of this course is to learn how to write a scientific text. To achieve this, at the end of this course you will: know what the main characteristics are of a scientific text be able to formulate a main question be able to find, critically read and manage scientific literature be able to use literature properly and avoid plagiarism be able to build up your argumentation be able to structure an article according to the conventions in your field of study be able to use scientific English style be able to use tables and figures to support and communicate your results be able to give feedback on somebody elses article be able to use feedback for improving your work.	
Education Method	(Online and/or on campus) practical, in 6 sessions (attendance mandatory). Every week you have to read some background information about scientific writing and hand in a part of your text. Participants must attend all sessions - one missed session is allowed only - and hand in all assignments in time. Students who receive a pass for this course are rewarded with 2 ects. This equals 56 hours of study. A total of 12 hours is spent on attending (online) classes, in which you can ask questions, discuss the feedback you received on your work and discuss aspects of scientific writing with your fellow students; the remaining 44 hours is for self study, writing and revising. In seven weeks, from preparing lecture 2 up to handing in your final article in week 8, you will have to spend at least 6 hours of self study on this course, every week. It is important that you make sure you have this time in your personal schedule. At the beginning of the course you will mainly be reading up about scientific writing and the subject of your text. As the course proceeds, you will be spending more of your time on writing, giving feedback and revising your own text.	
Books	Theory about academic writing will be made available through Brightspace.	
Assessment	You write a scientific article of 3000 words (excluding the list of references and the abstract, 10% margin in the word count). You have to hand in a parts of your article every week. Your final grade is based on the final article. An evaluation form for the grading of the article is available throughout the course.	
Elective	Yes	
Category	MSc level	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 2 2021

CS4015	Behaviour Change Support Systems	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	none	
Course Contents	Behavior change support systems (BCSS) are computer-based systems that support individuals to form, alter or reinforce cognitions, attitudes or behaviors without using coercion or deception. They can serve individuals throughout the various stages of a change process, such as awareness developing, contemplation, action strategy development, development of new behaviors, and maintaining these new behaviors. Virtual healthcare coaches, negotiation support systems, and applications that provide individuals with personalize financial guidance are three examples of these systems. To establish, modify or maintain change BCSS can deploy computerized persuasive strategies (e.g. reducing effort to establish target behavior, or argumentation and reflection strategies), simulations (e.g. serious gaming, virtual reality), relational software agents (e.g. ePartners, virtual coaches), and personalization based on longitudinal user data. BCSS are found in many domains, including education, sales, negotiation, management, and particular in the health domain.	
Study Goals	The course allows students to achieve understanding of principles, concepts and theories underlying BCCS systems and methods for designing them.	
Education Method	In the pre-recorded video material, theories, principles and methods are presented, discussed and illustrated with examples from the field. The video material is support by online self-tests. In the lectures, teacher and students discuss and make assignment related to pre-recorded material of that week. At home students work on their own in small groups on coursework assignments to develop a product design for a BCSS. In the practicum session student groups presented the progress on their coursework and receive feedback. Expected Workload Pre-recorded video material: 18 hours (2 hours × 9 lectures) lecture: 10 hours (10 × 1 hours) practicum 14 hours (7 × 2 hours) Reading time: 36 hours (9 × 4 hours reading time) Preparation presentation: 3 hours (3 × 1 hours for each presentation) Coursework project, including writing report, and final presentation: 50 hours Exam preparation and revision: 9 hours Total = 140 hours	
Literature and Study Materials	Will be announced on brightspace	
Books	Wendel, S. (2013). Designing for Behavior Change: Applying Psychology and Behavioral Economics. " O'Reilly Media, Inc.".	
Assessment	The course is assessed by coursework and an exam as follows: (60%) computerised examination (or oral exam) (40%) Coursework Project (resulting in a report, and final presentation include question and answer round where individual group members are assessed on coursework) If the expected number of students registering for exam or resit is small, the teacher might decide to replace the computerized examination by an oral examination. Separate marks will be given for exam and coursework, only a combined mark is recorded in Osiris. A passing final grade for the course can only be earned when for both the exam and coursework at least a 5.0 is earned, and the weighted grade for exam and coursework is at least a 5.8. Resit coursework A second submission of modified coursework is only allowed for coursework that received a fail mark (<5.8) for the first submission. Overall resit mark for coursework will be capped to 5.8. Note that individual marks for coursework or computerised exam (or oral exam) do not carry to the next year. disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	2	
Co-Instructor	M.L. Tielman	

CS4055	High Performance Data Networking	5
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Basic understanding of networking and programming (ideally Python).	
Course Contents	The Internet has become of critical importance to society. However, the large size of networks and abundance of protocols have made network management very complex. The novel concept of network programmability addresses this complexity and has resulted in a paradigm shift in how networks are (or can be) operated. The high-performance data networking course is an advanced networking course that will introduce you to the concept of network programmability and which treats fundamental networking concepts like Quality of Service and network resilience.	
Study Goals	The learning objectives of this course are twofold: (1) The student should gain knowledge of the treated networking technologies. (2) The student should be able to apply and work with the programmable network technologies in a network emulator (Mininet).	
Education Method	Approximately 50% of the course will consist of lectures and selfstudy and 50% focuses on (homework) exercises and instruction classes.	
Literature and Study Materials	Slides and a reader containing the exercise material.	
Assessment	The final assessment will be based on an exam that covers both the theory from the slides as well as the content from the reader.	
Remarks	Disclaimer: The information about CS4055 (including its assessment) may change depending on the developments around the coronavirus.	

CS4090	Quantum Communication and Cryptography	5
Responsible Instructor	Dr. S.D.C. Wehner	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear Algebra, Probability & Statistics, Q101 (Fundamentals of quantum information)	
Course Contents	This class will introduce you to the fascinating field of quantum communication! We will look at the state of the art of quantum networks, and explore techniques for building quantum repeaters that promise to deliver qubits over long distances. We also briefly look at one of the most famous application of quantum cryptography, quantum key distribution. Caution: 1. This class requires you to take "Fundamentals of Quantum Information" in Quarter 1 2. The focus of this class is presently on quantum communication, and we will only briefly look at quantum cryptography. As such, this class is not held in flipped classroom style in conjunction with edX QuCryptoX as in previous years.	
Study Goals	The student will acquire: A good understanding of the fundamental concepts of quantum communication Insight into the differences between classical and quantum communication and cryptography Skill set required to follow the remainder of the quantum curriculum (Q301 Quantum hardware and Q401 Quantum electronics)	
Education Method	Lectures and tutorials. If remote classes continue this fall, then recorded lectures and live discussion session	
Literature and Study Materials	Primary: Slides Review Articles Auxilliary: Nielsen and Chuang Quantum computation and information, Cambridge University Press.	
Assessment	Homework (70%) and Final Project (30%) disclaimer: information may change depending on the developments around the coronavirus.	
Tags	Abstract Adventurous Algorithmics Challenging Group Dynamics/Project Organisation Information & Communication Integrated Intensive Involved Lineair Algebra Mathematics Physics Quantum Signals Technology Telecommunication	

CS4200-B	Compiler Construction B	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/6/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	- CS4200-A: Compiler Construction A (recommended) - programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Note that the title of this course should be "Compiler Construction B", not "Compiler Construction Project". The course combines theory and practice of compiler back-ends. Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-B covers the following topics: * Transformation - rewrite rules, rewrite strategies - simplification, desugaring * Dynamic semantics and code generation - operational semantics, program execution - virtual machines, assembly code, byte code - code generation - memory management, garbage collection * Data-flow analysis - control-flow, data-flow - monotone frameworks, worklist algorithm	
Study Goals	After this course students students should be able to * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define transformations on abstract syntax terms to simplify programs * Define a code generator that translates source language abstract syntax trees to object language instructions using rewrite rules * Define data-flow analyses using control-flow and data-flow rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the back-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%)	

	- course project (50%) Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	not applicable
Judgement	to be decided

CS4220	**Machine Learning 1**	**5**
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

IN4089	Data Visualization	5
Responsible Instructor	T. Höllt	
Instructor	Prof.dr. E. Eisemann	
Contact Hours / Week x/x/x/x	0/2/0/0 & lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	Master course MKE/ST/DS	
Expected prior knowledge	IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required). The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.	
Course Contents	Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization, The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines. As a computer science course, affinity to algorithmic thinking and programing skills will be needed. Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.	
Study Goals	The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data. Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more. By the end of the course, you should be able to LO1: Discuss a large range of visualization techniques. LO2: Discuss a perception principle of visualization. LO3: Explain mathematical principles and algorithms of visualization techniques. LO4: Design suitable visualization systems for a given practical data analysis problem. LO5: Implement visualization systems for a given practical data analysis problem.	
Education Method	Lectures, practical assignments, self-study, and projects.	
Literature and Study Materials	Course slides, instructions for projects, and selected literature. Chapters from: Visualization Analysis and Design Author: Tamara Munzner CRC Press Visual Computing for Medicine 2nd Edition Theory, Algorithms, and Applications Authors: Bernhard Preim Charl Botha Morgan Kaufmann All available in electronic form via Brightspace or at TUDelft library.	
Assessment	The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Judgement	The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam. The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project. Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions. The project is evaluated based on the developed result, its documentation and presentation. Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.	

IN4150	Distributed Algorithms	6
Responsible Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Knowledge of Computer Networks (CSE1405) and Operating Systems (CSE2430) is useful as background for understanding this course.	
Course Contents	Introduction to distributed algorithms; notions of time and ordering of events; distributed algorithms for message ordering, detecting global states, termination detection, deadlock detection, mutual exclusion, election, minimum-weight spanning trees, fault tolerance, consensus, and agreement; blockchain technology and its relation with consensus.	
Study Goals	Understand the main fundamental problems in distributed systems Understand the most important distributed algorithms that solve these problems Be able to reason about the execution of distributed algorithms Be able to program distributed algorithms Be able to select and summarize relevant literature on distributed algorithms	
Education Method	Lectures and lab work executed in groups of two students	
Literature and Study Materials	Lecture notes and lecture slides (available on Brightspace)	
Assessment	One paper summary and an in-person, on-campus written exam (closed book), which together are an integrated whole. The grade of this exam, which includes the grade of the paper summary (with a weight of 25%), is the grade for the course. In case in-person, on-campus written exams are not possible because of COVID-19, the written exam will be replaced by individual online oral exams.	
Permitted Materials during Tests	None except the list of algorithms	
Remarks	Lab work is 40 hrs.	

IN4302TU	Building Serious Games	5
Responsible Instructor	Dr.ir. A.R. Bidarra	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2	
Course Language	English	
Expected prior knowledge	For CS students: programming experience with some object-oriented language; experience with graphics, AI and/or some game engine(s) is a plus. For all students: though not compulsory, it may be convenient to have followed the course SEN9235 (Game Design Project), which is taught in the first quarter.	
Course Contents	Project-based interdisciplinary course, open to MSc students of all faculties. The main goal of the project is to take students with varying talents, backgrounds, and perspectives and put them together to do what none of them could do alone: to design and implement a serious game aimed at being applied in a real-world setting (educational, social, training, health-related, etc.). The emphasis is both on constructively fulfilling the game requirements, and on deploying the adequate technology for that purpose. Assignments for this course will be provided by real-world end-users (e.g. companies or the Science Centre Delft), to whom the group will be reporting throughout the term of the project.	
Study Goals	At the end of the project, the student will demonstrate proficiency in the following aspects: o identifying and valuing the soft skills necessary to work in interdisciplinary teams o responsibly interacting within a team, integrating its members' varying talents and expertise o adapting with flexibility to the dynamic requirements of a complex external assignment o translating feedback received into proactive personal development steps Additionally, the CS student will demonstrate proficiency in the following specific aspects: o identifying, selecting and deploying the most adequate game technologies for the given serious game domain and constraints o deepening programming skills while building a complex and large software system in an agile context	
Education Method	Project: teams work intensively as a small game studio. Also a few plenary sessions and/or lectures (though less likely in 2021/22).	
Assessment	Project assessment will be based on a combination of: - (~50%) product grade: unique for the whole group, based on both the game itself and the required documentation; - (~45%) process grade (individual), including personal contribution, performance, attitude, and peer evaluation; - (5%) final presentation. The commissioner will be involved both as advisor and as assessor. The final documentation will include writing a scientific paper and actually submitting it to a conference on serious games and/or their application. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr. R. Marroquim	

IN4341	Performance Analysis	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	0/2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	This course applies probability theory and the theory of stochastic processes to the design and performance evaluation of complex networks such as man-made networks as telecommunication, computer and embedded networks and biological networks. The computation with random variables is reviewed. Markov processes and queuing theory will be introduced to the current important concept of "Quality of Service (QoS)" provisioning and to the computation of the blocking probabilities in telephony (both fixed as mobile). Several applications (e.g. the robustness of networks, epidemics in networks, the Internet shortest path routing) are also included. More details are found on brightspace.	
Study Goals	The course intends to provide students with mathematical techniques, in particular probabilistic methods and graph theory, to compare the performance of different network designs and protocols.	
Education Method	Lectures and homework after each class	
Literature and Study Materials	We follow the book Performance Analysis of Complex Networks and Systems, by P. Van Mieghem, Cambridge University Press (2014). See http://www.nas.ewi.tudelft.nl/people/Piet/bookPA.html	
Assessment	Written and closed book. A formularium is provided that can be consulted at the examination. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 3 2021

AP3132	Advanced Digital Image Processing	6
Responsible Instructor	Prof.dr. B. Rieger	
Instructor	Dr. F.M. Vos	
Contact Hours / Week x/x/x/x	0/0/4/2	
Education Period	3 4	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Basics of signal processing, image processing, linear algebra, elementary statistics.	
Course Contents	The course Advanced Digital Image Processing covers the principles of several state-of-art image processing techniques. Particularly, students will study the theory of sophisticated algorithms for: 1. Multi-resolution Image Processing: gaussian scale space, windowed Fourier transform, Gabor filters, multi-resolution systems (pyramids, subband coding and Haar transform), multi-resolution expansions (scaling functions and wavelet functions), wavelet Transforms (Wave series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform (CWT), Fast Wavelet Transform (FWT)); 2. Morpological Image Processing: advanced operations for binary morphology; definitions of gray-scale morphology regarding erosion, dilation, opening, closing; application of gray-scale morphology including smoothing, gradient, second derivatives (top hat) and morphological sieves (granulometry); 3. Image Feature Representation and Description: measurement principles: accuracy vs. precision ; size measurements: area and length (perimeter); shape descriptors of the object outline: form factor, sphericity, eccentricity, curvature signature, bending energy, Fourier descriptors, convex hull, topology; shape descriptors of the gray-scale object: moments, PCA, intensity and density; structure tensor in 2D and 3D: Harris Stephens corner detector, isophote curvature. 4. Motion and optic flow: Taylor expansion method; dual and multi-frame image registration, optic flow; 5. Image Restoration: Noise filtering, Wiener filtering, inverse filtering, geometric transformation, grey value interpolation; 6. Image Segmentation: thresholding, edge and contour detection, data-driven segmentation (boundary detection, region-based segmentation, watersheds, graph-cut, meean shift), model-driven image segmentation (Hough transform, template matching, deformable templates, active contours, ASM/AAM, level sets).	
Study Goals	General learning objectives of the course are: 1. Student has knowledge of can explain the function of state-of-the-art image processing algorithms; 2. Student can solve elementary problems in image processing using Python/MATLAB? programming; 3. Student can solve more advanced problems without implementation, but sketching steps towards a solution; 4. Student can independently acquire new knowledge about image processing from the current literature and present and report about it.	
Education Method	Lectures, practicals and group assignment with plenary presentation and discussion.	
Computer Use	Matlab including the dipimage toolbox and/or other image processing toolboxes.	
Literature and Study Materials	Book 'Digital Image Processing', van R.C. Gonzalez en R.E. Woods, third edition, 2002, ISBN 9780131687288. (Online) Book 'Computer Vision, Algorithms and Applications', R. Szeliski, (http://szeliski.org/Book/). The online version is available for free. We have used the Book Introductory Techniques for 3-D Computer Vision, E. Trucco and A. Verri, ISBN 0-13-261108-2 in the past. Lecture notes Fundamentals of Image Processing (http://homepage.tudelft.nl/e3q6n/education/et4085/sheets/ppt/FIP2.2.pdf) PDF-files of the lecture slides (see Brightspace).	
Assessment	Closed book written exam and assignment. Both parts should be graded 5.8 or higher. A bonus point of 1.5 (to the exam) can be obtained by attending the practicals with 6 out of 8 passed. The final grade is the average of the two parts. The formula for the final grade is: ((0.85*EX + 0.15)+ AS)/ 2 or without the bonus point from the practicals: (EX+AS)/2 With EX the exam grade and AS the grade for the assignment. If you have not passed the exam or the resit, you will need to redo the assignment again next year!	
Permitted Materials during Tests	Closed book exam; books, print-out of pdf files of the lecture slides and lecture notes are not permitted during the written examination.	
Elective	Yes	
Tags	Image processing Matlab Physics	

CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
Responsible Instructor	Ir. S.E. Verwer	
Responsible Instructor	Dr. A. Panichella	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software is one of the most complex artifacts of mankind has ever created, but complexity is the enemy of correctness. Modern software testing and validation tools use a multitude of techniques geared toward correct computer code, most of these are base on artificial intelligence. In this course, we study these techniques in details, specifically we will understand and implement: Execution monitoring and taint analysis Branch distance computation Hill-climbing and genetic algorithms Concrete and symbolic (concolic) execution Active state machine learning Genetic programming The goal is to better understand and test software using artificial intelligence. Using the taught techniques you will be able to automatically: Discover which code is reachable Find (security) bugs in software Write tests that cover all reachable code Reverse engineer a code's functionality Patch code to remove bugs and failing tests	
Study Goals	The student will: Understand modern AI techniques for software testing. Be able to implement several such techniques from scratch: - smart fuzzing (probing software with input to find crashes/bugs), - symbolic execution (using logic to construct inputs that trigger specific code branches), - fault localization (given that a program fails, find the line of code responsible for the failure), and - automated program repair (using a patch library and genetic programming to improve code) Be able to apply this technology to locate bugs in real-world software implementations.	
Education Method	The main part of the course will consist of 3 lab assignments covering the theory (fuzzing&tainting, symbolic execution, automated program repair), and one lab assignment for the application to real software. The students will implement the taught techniques from scratch in the first 3 assignments, which will be scored with a pass/fail. All three assignments need to be passed to complete the course. The final lab will contain a recap from the first three assignments and an application of a state-of-the-art tool on real software. The final lab will be graded and be the final course grade. There will be instruction sessions where students can work on their assignment and ask the teachers for assistance.	
Assessment	First three lab assignments (pass/fail). Final lab (100%).	
Tags	Artificial intelligence Software	

CS4135	Software Verification	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Ir. K.F. Wullaert	
Contact Hours / Week x/x/x/x	0/0/2/0 + 0/0/4/0 practicum	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	How can we ensure that software cannot crash and is guaranteed to be correct? In this course we tackle this question by viewing programs and programming languages as mathematical objects. That way we can use logic to prove properties about programs and thereby guarantee that software is correct. To make reasoning about actual programs and programming languages feasible, we will not be doing these proofs by hand, but instead use a tool called a proof assistant to build proofs that can be checked by a computer. As we will show during this course, proof assistants turn the activity of doing proofs into programming. This course assumes familiarity with functional programming and elementary logic. This course is a specialization course for programming languages and software engineering	
Study Goals	After this course students will be able to: - State and prove properties of functional programs in logic. - Specify the semantics of a programming language in logic. - State and prove the correctness of imperative programs. - Use a proof assistant to perform a mechanized proof.	
Education Method	This course consists of a weekly lecture of 2 hours and a lab session of 4 hours. During the lab sessions students will work on proving simple theorems. Towards the end of the course students will carry out research projects that apply the ideas of the course.	
Literature and Study Materials	Supplementary material: Free online text book "Logic and Proof": https://leanprover.github.io/logic_and_proof/ Free online text book "The Hitchhikers Guide to Logical Verification": https://github.com/blanchette/logical_verification_2021/raw/main/hitchhikers_guide.pdf	
Assessment	The final grade consists of the following parts: - A programming project in a proof assistant. - A written exam Both have weight 50% and both should be 5 or higher. The weighted average should be 5.8 or higher. The research project should be done individually.	
Co-Instructor	E. Visser	

CS4160	Blockchain Engineering	5
Responsible Instructor	Dr.ir. J.A. Pouwelse	
Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	In this course you will learn all aspects of blockchain technology, including tamper-proof data structures, digital identities, transitive trust, fault tolerance, distributed consensus, smart contracts and applications. Ledgers and blockchains are an emerging technology with the potential to radically improve financial transactions, supply-chain flows, transactions in general, and distributed databases. The first three weeks of the course will provide a fast-paced introduction to Bitcoin, Ethereum, and TrustChain developed by TUDelft itself. The main component in this course is a team-based complex engineering project. This course is designed for computer scientists to understand blockchain technology and to produce significant hands-on experience. To provide a deep understanding of blockchain technology and understand why it is special you need to experience first-hand how it operates at a detailed technical level. Students design, implement, and test their own independent project in teams of 3-5 students. Students can choose from a pool of possible project ideas. This course requires you to like software engineering. Topics covered: -Blockchain basics and evolution Bitcoin 1st generation, smart contract generation, future 3rd generation (trust or trust in math) -identity and transitive trust Authentication and security primitives, tamper-proof identities, trust models, MITM attacks, Sybil attacks, and TrustChain by TUDelft -Consensus models Proof-of-work, permissioned, Proof-of-stake, Corda no-global-consensus, TUDelft bottom-up fast consensus model -Smart Contract pro/con encrypted data, Bitcoin scripts, Etherium execution model, Hyperledger + Docker issues, Corda Jar file approach, Tezos difficult to use, powerful technology, vision of the future: trusted verified execution -Markets and exchanges Disruption by open markets, winner-takes-all, and multi-sided market platforms, Uber, Airbnb, 22 years of eBay, Silk Road, honesty among drug dealers, the role of trust in markets, P2P exchange markets -Decentralized Autonomous Organization, novel method to collaborate and organise any economic activity Within this 2021 edition "the Delft DAO" will be prominently featured. TUDelft achieved a world-first in DAO research. We devised a full end-to-end proof-of-principle of a DAO which is capable of 0) near unbounded scalability 1) controlling money 2) democratic decision making and 3) continuous sustained self-evolution. This course provides you with the knowledge to work with this advanced technology. After this course you will have a firm grasp on the current operational blockchain-based systems, realistic view of real-world applications that may be built on top of ledger technology. You will be able to reason and discuss the open challenges and questions that still need to be resolved. This course is a key course for distributed systems students.	
Study Goals	After this course students are able to design and engineer complex blockchain-based systems. Students are able to describe blockchain technology, the various consensus model, smart contracts, markets, and relation to existing database technology. Student are able to setup a new architecture for blockchain applications.	
Education Method	This course consists of four 2-hour lectures. Each lecture is followed by a 4-hour homework period in the same week focused on understanding the background material. In week 1 you will form teams and initiate work on your blockchain engineering project. A list of projects to select from will be provided at the start of this course.	
Literature and Study Materials	Online course textbook: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Narayanan, Bonneau, Felten, Miller and Goldfeder.	
Prerequisites	It is highly recommended to follow this course (see remarks): Security and Cryptography (Q1) Distributed Algorithms (Q2)	
Assessment	The final grade reflects the quality of your work and team cooperation. This course has a minimal amount of formalities. You will write down your project results in a single-page report, IEEE style. You will be graded on your open source efforts located on Github and single-page report. Your grade will be expressed on a scale of 0 to 10. (resits or repair options are not offered for this course) Covid-19 disclaimer: the assessment and course format could be altered at any time !!!	
Remarks	This class has a limited capacity (50). If there is a larger number of enrollments than the capacity of the class, students will be assigned to their preferred blockchain engineering project based on their background, engineering experience level, and match to the course goals. Students who followed Security and Cryptography (Q1) and are also enrolled in Distributed Algorithms (Q2) will have priority for placement. Mathematics students are exempts from this, if they can show some minimal software development experience (e.g. Github profile). Finally, students with a Grade Point Average of 8.0 or higher are eligible for the challenging scientific projects, resulting in a research paper. These project receive intense guidance, but have no capacity limits.	

CS4195	Modeling and Data Analysis in Complex Networks	5
Responsible Instructor	H. Wang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The assignment and final project require basic programming skill.	
Course Contents	Big Data is mostly obtained from features of components and the interactions among components in large complex systems. Examples are (1) end user features and interactions in both online and real-world social networks like Twitter, LinkedIn (2) data from content sharing platforms such as YouTube (3) physiological data of the brain and (4) stock prices etc. in economic systems. Such a dataset is networked in nature i.e. the data of the system components or interactions are (cor)related to each other. This course introduces the basic methodologies to analyze, model, interpret and possibly to predict such Networked Data, combining advances from network science, modeling of dynamic processes and statistical physics, beyond machine learning algorithms. These methods will be applied to diverse real-world datasets obtained from e.g. Facebook, LinkedIn, YouTube, the brain etc.	
Study Goals	After this course, students could construct a network based on the dataset, characterize and model the network in order to e.g. detect patterns and anomalies, model the data via dynamic processes (e.g. viral spreading) on networks to decode the underlying governing mechanisms of e.g. information/error/behavior contagion and to predict e.g. the popularity of a product, news, disease, computer virus, control the contagion process such as maximize the information prevalence and market share. Students could obtain an overview of the Msc/Phd projects on the frontiers of networked data analysis.	
Education Method	In total, there will be about 7 lectures. Students will also learn via an assignment and a final project (each group gets individual supervision).	
Assessment	The final grade is based on the assignment (20%) and final project (80%). There is no resit for both the assignment and the project. disclaimer: information may change depending on the developments around the coronavirus.	

CS4210-A	Algorithms for Intelligent Decision Making	5
Responsible Instructor	Dr. N. Yorke-Smith	
Instructor	Dr. E. Demirovi	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Recommended: IN4010: Artificial Intelligence Techniques, or equivalent; and/or IN4301: Advanced Algorithms, or equivalent Required: basic course(s) in algorithm design and analysis, logic and probability; basic programming (in Python)	
Course Contents	Decision making is at the centre of artificial intelligence. This course gives you practical skills on a solid theoretical base. The course looks at solving mathematical models of NP-hard discrete optimisation problems. These kinds of problems lie at the heart of AI techniques such as planning, machine learning and mechanism design, and more generally combinatorial optimisation. You will learn about a range of modelling techniques from boolean satisfiability to constraint programming, and how advanced solvers for these models work. The course has plenty of real-world case studies as well as theoretical results. Apply the skills you learn in this course by taking CS4210-B: Intelligent Decision Making Project in quarter 4!	
Study Goals	By the end of this course, you will be able to identify features of real-world combinatorial decision problems, and be able to model and design systems for simplified instances of these problems using boolean satisfiability, mixed integer programming, and constraint programming over finite and real domains. You will be able to explain how SAT, CP and LCG solvers work in some detail, and how MIP solvers work at a high level.	
Education Method	Lectures, homework exercises (optional), and programming assignments. The expected workload is: 30% lectures (including preparation for the exams) 40% homework exercises (optional) 30% programming assignments	
Literature and Study Materials	Provided on Brightspace	
Assessment	The final grade depends on the grades obtained for (a) programming assignments (2 in total) [30%] and (b) the exam [70%]. The final grade is computed from the unrounded grades for the components. The final grade for the programming assignment is a uniformally-weighted average of the unrounded grades obtained for the two assignments. Programming assignments can be completed by two students working together. The exam is graded on a scale from 1 to 10. A resit will be available for the exam. The result for the exam is determined by the maximum score obtained for the original exam and the resit. In order to pass the course, the rounded grade (after resit if applicable) for each part of the course must be at least 5.0, and the rounded final grade on the course must be at least 5.8. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Algorithmics Artificial intelligence Group work Modelling Optimalisation Programming Projects Small groups	

CS4225	Educational Technologies	5
Responsible Instructor	Prof. M.M. Specht	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	* Theories of Human Information Processing and Learning * Learning Management Systems * Learning Analytics * Personalisation and Adaptive Educational Systems * Mobile and Seamless Learning Technologies * Artificial Intelligence in Education * Realtime Learning Technologies * Project Design * Project Implementation	
Study Goals	The course will enable you to classify, understand, design and implement the core functionalities and systems for supporting human learning processes. As well current practices implemented as also approaches for technology enhanced learning currently researched will be presented. You will learn how educational technologies provide human learning process support, implement guidance and recommendation, create personalised learning support, as also give real-time feedback and support reflection of learners. In the final project you will identify a problem, design a solution based on the presented approaches and implement your own educational technology solution.	
Education Method	Lectures, weekly assignments and quiz questions, final project	
Assessment	Weekly assignments 30%, Final project 70% disclaimer: information may change depending on the developments around the coronavirus.	

CS4230	Machine Learning 2	5
Responsible Instructor	M. Loog	
Instructor	Dr.ir. J.H. Krijthe	
Instructor	Dr. F.A. Oliehoek	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/4/4	
Education Period	3 4	
Start Education	3	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course is the more advanced and research oriented follow-up to CS4220 [Machine Learning 1]. The content of the latter is, therefore, expected as prior knowledge.	
Course Contents	The course will treat a number of machine learning theories and techniques in detail and on an advanced level. Possible topics : - learning theory - Bayesian networks - online learning - Rademacher complexity - Markov decision processes - semi-supervised learning - reinforcement learning - active learning - causal reasoning and discovery	
Study Goals	After successfully completing the course, the student is able to apply the techniques and theories that have been covered in the course. In addition, they are able to develop learning strategies for new and previously unseen situations. Moreover, the student can provide reasoned justifications for these strategies based, for instance, on theory and/or experiment.	
Education Method	Lectures + Q&A sessions	
Assessment	Grading is based on two parts. Following the lectures -- we have about 11 of those, there is an individual assignment that will be graded pass/fail. In addition, there is a written examination that will be graded on a scale from 1 to 10. You pass the course when you both have a pass for the assignment and a passing grade for the written exam. Upon passing the course, your final grade will be the grade for the exam. Finally, note that there is a resit for the written examination, but not for the report. disclaimer: information may change depending on the developments around the coronavirus.	

CS4235	Socio-Cognitive Engineering	5
Responsible Instructor	Prof.dr. M.A. Neerincx	
Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Basic prior knowledge on human-computer interaction is helpful, but not required.	
Course Contents	Whether you are playing a game in virtual reality, driving a semi-autonomous car, educating yourself in a MOOC, or harmonizing your health and lifestyle via a mobile app; nowadays intelligent networked information and communication technology is omnipresent. This course focuses on the design of human-aware intelligence into such environments, to support joint human-technology performances that bring about positive human experiences (such as social robots that help activity coordinators guide health-promoting games for people with dementia, http://rejam.tudelft.nl). In the Socio-Cognitive Engineering (SCE) course (MSc level), you will become acquainted with the application of a coherent set of methods for the design and evaluation of human--agent collaboration. Based on the SCE-method, we will elaborate on the state of the art of intelligent user interfaces (ePartners), such as artificial personal assistants, artificial team mates, eCoaches, social intelligence, and companion agents. The main topics of study are: - Design methods: Cognitive Engineering, Value Sensitive Design, Scenario-based Design, Claims Analysis, Design Rationale, Design Patterns. - Design for collective intelligence: Knowledge Representation, Ontology Engineering, Mental Models, Theory of Mind, ePartners, Adaptive Automation, Socially Intelligent Interfaces. - Design Evaluation: Prototyping, Test Methods, Measures, Questionnaires, Ethics. - Human Factors Theories and Models: Human Cognition & Learning, Memory, Emotion, Task Load, Human-Agent Teamwork, Behavior Change and Persuasive Technology.	
Study Goals	At the end of this course, students will be able to: 1. Explain the essential concepts of the design methods addressed in the course. 2. Explain the (dis)advantages of various design methods and their complementarity. 3. Apply the design methods addressed in the course in their research and design projects. 4. Explain what a design rationale is. 5. Construct a design rationale. 6. Create design specifications that are grounded in a design rationale. 7. Evaluate the strengths and weaknesses of a design rationale, e.g. using human-centered evaluations that test the design rationale. 8. Explain some of the state of the art human factors theories, models, and methods relevant to intelligent user interfaces, human-agent collaboration, and ePartner technology. 9. Write a structured report about a design-test cycle, with sufficient detail for a new group of researchers to continue the research. 10. Present work on a design project to an academic audience. 11. Work in a group on collaborative assignments.	
Education Method	LECTURES During the lectures, the teachers will present a range of theories, models, and methods relevant to socio-cognitive engineering. Students are required to read a number of scientific papers which are made available on Brightspace, along with the sheets/slides of the lectures. Together, the sheets/slides and the papers provide the students with the required theoretical knowledge to work on the practical project, and to learn about relevant design methods, human factors theories, conceptual solutions, and design principles. Most of the lectures include practical assignments and discussions stimulating the students to apply the contents of the lecture to their project (also see Project). PROJECT In the project, students work in groups to apply the knowledge acquired during the lectures. Students are required to plan, execute, present, and report on a complete design cycle (i.e. design, prototype, and evaluation) for a given design problem. This year (like the past years), the design problem is a social robot for older adults with dementia, and their social environment (https://rejam.tudelft.nl). The objective of the social robot is to improve humans physical, social, cognitive, and emotional well-being. The students will use the Wiki Socio-Cognitive Engineering (WiSCE) tool to specify the design rationale and its evaluation, step-by-step (see also https://scetool.ewi.tudelft.nl). Throughout the course, students will give presentations about their progress, on the design and evaluation of their prototype.	
Literature and Study Materials	Papers from scientific journals on Brightspace. Lecture notes on Brightspace.	
Assessment	Literature and study material consist of: - Papers from scientific journals on Brightspace. - Lecture notes on Brightspace The module assessment concerns the processing and application of the theory and methods; the construction of the design (rationale) and the evaluation; and the provision of the resulting concise and coherent report (including the lessons learned): Presentations (10%) Prototype (10%) Project report according to the prescribed format (70%) Individual reflection (10%)	
Exam Hours	There is no exam. The assessment is based on a paper, presentation and report. During the course, students will receive feedback on interim work. There is no resit after the end of the course.	

CS4240	Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	Seminar Computer Vision by Deep Learning	
Expected prior knowledge	Basic pattern Recognition, Basic Machine learning, Basic statistics, Basic probability theory, Programming experience (python + numpy). Note: Without some affinity/experience with python (and numpy) programming it might still be possible to pass the course, but will become quite difficult.	
Course Contents	In this course we will look at a specific field of Artificial Intelligence and Machine Learning: Deep learning. Deep learning has shown remarkable success with large data sets and unstructured input data such as raw images/audio/text. Topics include: feed forward networks, back-propagation, optimization, convolutional nets, recurrent nets, self-attention, unsupervised methods. The course will have lectures, a seminar, a lab practical and a project: - The lectures will be on generic topics, following the book; building the backbone. - The lab assignments will have you apply basic concepts of the lecture in python notebooks. - The seminar will have students read, critique, and present relevant deep learning research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The project will have students apply and design their own (small) deep learning project in the context of scientific reproduction.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the different deep learning techniques reviewed in the course, such as SGD, MLPs, CNNs, RNNs, GANs. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [LO4]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (eg: Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) problem domain which may need to re-implement, run, evaluate, investigate, extend existing research or code [LO8] Write clearly and concisely about your code, method, results, and analysis.	
Education Method	Lectures for basic theory based on the literature Assignments: we help you to become familiar with PyTorch; applying concepts from the lecture on small problems. Lab project: design and execute your own deep learning project in the context of scientific reproductions using https://reproducedpapers.org/ Seminar: paper reading, critiquing, and presenting.	
Literature and Study Materials	Books: freely available online: - http://www.deeplearningbook.org/ - https://d2l.ai/ Research papers will be made available through Brightspace. Assignments are based on PyTorch: https://pytorch.org/	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Project: in a small group of students you work on a deep learning paper reproducibility project. 3. Exam about the papers, assignments, and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4275	Web Programming Languages	5
Responsible Instructor	E. Visser	
Instructor	Dr.ir. D.M. Groenewegen	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	Languages and frameworks for web programming are constantly evolving. Over the past decade, there has been a large shift from applications with server-side rendering of separate web pages, to single page applications with client-side rendering and web service back-ends. One of the strengths of web programming technologies is separation of concerns. The concerns such as describing content, styling, behavior, and persistence, are often separated with their own domain-specific languages. A particular programming problem that newer web programming languages tackle is dynamic user interfaces with automatic fine-grained updates. This problem is not restricted to web applications, but applies to any GUI programming abstraction. Consequently, the technologies for web programming are also relevant for development of cross-platform mobile and desktop applications. In this course, students will analyze web languages and frameworks from a programming language perspective. They will explore the underlying concepts and abstractions, and learn how the tools relate to each other. The investigated web technologies range from more traditional full-stack web development solutions with persistence and templating, to popular client-side UI solutions with fine-grained updates and state synchronization.	
Study Goals	The course gives students the conceptual and technical skills to understand the role of programming languages in web programming and the advantages and disadvantages of different approaches.	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab sessions - reading lecture material and papers - making project assignments	
Assessment	Students get a grade for each of the project assignments. The final grade is the weighted average of the grades for the project assignments. There will not be a resit for the course. Disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is the average of the grades for the project assignments.	

CS4400	Deep Reinforcement Learning	5
Responsible Instructor	Dr. J.W. Böhmer	
Instructor	Prof.dr. M.T.J. Spaan	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students must have passed IN4010(-12) "Artificial Intelligence Techniques", or have acquired equivalent knowledge about: - basic probability theory, analysis and algebra - general machine learning methodology, e.g. regression - fully and partially observable Markov decision processes - tabular reinforcement learning methods, e.g. Q-learning - the exploration/exploitation trade-off, e.g. RMAX or UCB - multi-agent learning, e.g. centralized training and decentralized execution Students are encouraged to close any gaps in the above knowledge and to familiarize themselves with the Python/PyTorch deep-learning framework before the start of the course.	
Course Contents	This course will cover the breadth of modern model-free RL methods, discuss their limitations and introduce a variety of current research topics. In particular, we expect to cover the following: - deep learning methodology and architectures - stabilization of approximated value estimation - modern actor-critic methods - planning as inference - exploration with deep networks - offline reinforcement learning - deep multi-agent reinforcement learning - multi-task and meta learning	
Study Goals	After successful completion of this course, students - can list the strengths and limitations of modern deep RL approaches, - explain the underlying concepts of the discussed methods, and how they differ from each other, - can implement selected algorithms/architectures, and - can analyze a new task to decide which algorithms/architectures to apply.	
Education Method	The course will be taught in one lecture per week and the content will be solidified in homework, which will be presented in one mandatory tutorial per week.	
Assessment	The final grade will be 100% determined by a written exam at the end of Q3, with a resit possibility in Q4. To be eligible for the exam, students must attend weekly tutorials and hand in homework exercises. Homework will not be individually graded, but at least 75% of the answers must be of sufficient quality (in terms of time commitment, not necessarily correctness) to be eligible to take the exam.	
Maximum number of participants	As this is the first time the course will be taught, it will be restricted to 30 participants.	

CS4405	Analysis of Concurrent and Distributed Programs	5
Responsible Instructor	Dr. B. Özkan	
Responsible Instructor	Dr. S.S. Chakraborty	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software systems are becoming highly concurrent and distributed to utilize modern multicore architectures and increasing speed and bandwidth in networks. Shared-memory concurrency in multicore programs and message-passing concurrency in distributed programs share many common abstractions and problems. In the multicore era, all performance-critical software employs some form of concurrent programming; typically shared memory concurrency. In this setting, programmers use a number of primitives to develop efficient and correct concurrent programs. To do so the programmers have to understand the behaviors of the primitives and reason about them. It is also important to match the programming paradigms and underlying architectures. For instance, traditionally programmers have assumed that a multithreaded program executed simply by interleaving the executions of its threadsa model known as sequential consistency (SC). This assumption is, however, invalidated both by mainstream multicore architectures, which often execute instructions out of order, and by compilers, whose optimizations affect the outcomes of concurrent programs. As a result, concurrent programs have more outcomes than SC allows. In the distributed setting, the units of concurrency are independent processes that do not share memory but communicate by exchanging asynchronous messages. The execution of such a system involves two main sources of nondeterminism: concurrency and partial failures. As the processes run concurrently, the exchanged messages can be delivered and processed in many different orderings. The distributed set of processes is also prone to network of process failures. The trade-off between the systems availability in the existence of failures and the consistency between the processes gives rise to a spectrum of weak consistency notions. It is important to reason about concurrency, possible failures, and consistency guarantees to implement distributed programs correctly and understand their behavior. This course aims to explore analysis techniques for concurrent and distributed programs. Outline of Lectures: Shared memory concurrency: - Abstractions for shared memory concurrency - Relaxed memory concurrency - Correctness of concurrent programs Distributed concurrency: - Distributed system components, models and assumptions - Fundamental abstractions for distributed systems	
Study Goals	This course aims to give students a deep understanding of concurrency and distribution in modern systems and hands-on experience for analyzing these systems. At the end of the course, the students will be able to: - Analyze and reason about concurrent and distributed programs - Apply and analyze existing techniques on unseen problems - Be able to pursue independent further research in the area	
Education Method	The course consists of the following education methods: - Lectures for reviewing concurrency and distribution concepts - Homeworks/assignments - Developing a course project, writing a report, and presenting it (course project) To finish the course, students (in teams) will have to: - Study several papers which will be discussed during the lectures - Deliver their assignments - Deliver and present their implementation project	
Assessment	The final grade is composed of: research project implementation) (40%) + research project report (20%) + research project presentation (20%) + homework assignment (10% + 10%). No written exam. Resits are not offered.	

CS4415	Sustainable Software Engineering	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	L. Miranda da Cruz	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	Sustainable Software Engineering is an overarching discipline that addresses the long-term consequences of designing, building, and releasing a software project. By definition, sustainability covers five main perspectives: environmental, social, individual, economic, technical. This course mainly focuses on the first, also known as Green Software Engineering. Incidentally, we will also cover some fundamental aspects of social and individual sustainability of software projects. Software Engineering (SE) has long addressed sustainability by narrowing it down to economic and technical sustainability. However, our society is facing major sustainability challenges that can no longer be overlooked by software engineers and computer scientists. It was estimated that, by 2040, the ICT sector will contribute to 14% of the global carbon footprint. Hence, environmental, social, and individual ought to be part of the equation when it comes to design, build, and release software systems. The problem is far from simple, but we need expert computer scientists to bring sustainability into the core values of the next generation of tech-leading organizations.	
Study Goals	After attending this course, you will be able to: LO1. Measure software energy consumption. LO2. Automate carbon-awareness in software development. LO3. Discuss sustainability principles. LO4. Solve sustainability issues in real software projects. LO5. Propose innovative strategies to monitor software sustainability.	
Education Method	To meet these objectives, you will be involved in a broad set of learning activities: lectures, paper reading, software analysis, software development, essay writing and presentation. These heterogenous set of activities aims at building a strong set of hard skills for energy-efficient code development combined with a strong set of soft-skills and critical thinking. Ideally, you will work on projects that will also help real-world software projects embrace a green software culture.	
Assessment	The assessment will be performed as part of the group project. It will include several steering meetings, an essay, a software repository, and a final presentation.	

CS4430	Network Security	5
Responsible Instructor	Dr.ing. A. Zarras	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	The course provides an overview of the most important concepts, methods, and best practices in computer and network security. In this course, students will obtain the knowledge and hands-on experience to secure networking and communication systems. The course's primary focus will be on technologies, protocols, attacks, and defenses. More precisely, starting from a review of common vulnerabilities and attack scenarios, the course will discuss the fundamentals of security engineering and their application in system design, review tools and methods to assess and test communication infrastructure from a security perspective. As a result, students will gain theoretical knowledge and hands-on experience in network attacks and defense methods. Knowledge activation and the transfer from conceptual understanding towards practical experience will be further facilitated by students implementing their own attack or defense tools on selected topics, as well as conducting measurements on the effectiveness of attack and defense schemes.	
Study Goals	See course contents.	
Education Method	Lectures, Labs, and Project.	
Assessment	Assignments and Project.	

EE4560	Information Theory	5
Responsible Instructor	Dr. J.A. Martinez Castaneda	
Instructor	G. Joseph	
Instructor	Dr.ir. J.H. Weber	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	This course explains the basic ideas of information theory and the correspondences between the elements of this theory and certain natural concepts of importance in a wide number of fields, such as transmission, storage, authoring and protection of data. On the basis of simple concepts from probabiliby calculus, models are developed for a discrete information source and a discrete communication channel. Further, the theoretical basics for developing source coding algorithms is provided, as well as the basics of optimal data transmission through a discrete communication channel. The following topics will be covered: * (Differential) Entropy, Relative Entropy and Mutual Information * Asymptotic Equipartition Property * Data Compression * Channel Capacity * Gaussian Channel * Rate-Distortion Theory * Network Information Theory	
Study Goals	Upon completion of this course the student will understand the fundamentals of Information Theory, which includes the following: (a) the correspondences between the elements of this theory and certain natural concepts of importance in a wide number of fields, such as transmission, storage, authoring and protection of data, (b) core theorems of information theory, (c) the models that are developed for a discrete information source and a discrete communication channel on the basis of simple concepts from probability calculus, (d) how to develop source coding algorithms, and (e) how to secure optimal data transmission through a (noisy) discrete communication channel.	
Education Method	lectures + mini project	
Assessment	CoVid-19 disclaimer: In light of the Corona crisis a remote assessment format could be implemented. Examination: Project and Exam The grade is determined by a project score (20%) and an exam score (80%). There are two opportunities for both. These are further explained below. Please note the exam format will depend on current CoViD-19 regulations. Project: The project is individual. Detailed instructions are listed in Brightspace, the project report is to be delivered via Brightspace. Exam: If regulation allows standard written examination on campus will be applied, otherwise we will use YouSeeU (Virtual Classroom) which is embedded in Brightspace. All details regarding the examination are listed in Brightspace. Grading: First opportunity: The project report should be submitted before the deadline (listed in Brightspace). The project score P1 is an integer between 0 and 10, while the exam score E1 is between 1 and 10 with a half-integer accuracy. The total weighted score is then (4×E1+P1)/5, which is rounded to the nearest grade in the set {5.0, 5.5, 6.0, 6.5, , 9.0, 9.5, 10.0} if both E1 and P1 are at least equal to 5. In case one or both are below 5, then the total weighted score is min(5.5,(4×E1+P1)/5), which is rounded to the nearest grade in the set {1.0, 1.5, 2.0, 2.5, , 5.0, 5.5}. In other words, a necessary condition to pass the course is that both the project score and the exam score must be at least equal to five. N.B.1: If the project report is not sent before the deadline, then P1=0. N.B.2: If the student does not participate in the exam, then E1=0. N.B.3: If the student already did an ee4560 project in a previous study year, then the student can request one of the lecturers before the project deadline by e-mail to let this be taken into account; this can be done in two ways: either to let the score of that project count as P1, or to take a different project from the list and to submit the report before the deadline, in which case P1 will be the maximum of the old and the new score. Second opportunity: A student not passing in the first opportunity or willing to improve his/her grade can redo the project, redo the exam, or both. In case a new project is done, the topic should be different. The project report should be submitted before the resit deadline (listed on Brightspace). A resit exam will be announced in Brightspace. With the project and/or exam scores being P2 and E2, the new total weighted score becomes (max{P1,P2}+4×max{E1,E2})/5, which is rounded as indicated above, still requiring that both the project score max{P1,P2}and the exam score max{E1,E2} must be at least equal to five. disclaimer: information may change depending on the developments around the coronavirus.	

ET4394	Wireless IoT and Local Area Networks	5
Responsible Instructor	Dr. P. Pawelczak	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Students are advised to follow the course Wireless Communications (ET4358) before taking this Wireless Networking course. An advantage is to have entry-level programming skills (Matlab, Python, C/C++). Nonetheless, students with little knowledge of programming will be helped.	
Course Contents	DISCLAIMER: this study guide information may change depending on the developments around the corona virus. The following modules will be discussed during the lectures: Introduction (example topics): - What is wireless networking - Where to search for (academic) wireless network literature and resources Medium Access Control (example topics): - WiFi: hidden/exposed terminal problem, Carrier Sense Multiple Access - Bluetooth standard: in-depth look into the channel hopping, protocol specifications WiFi (example topics): - Review of IEEE 802.11 standards - Protocol format - ISM band regulation - Adaptive Modulation and Coding - WiFi Matlab class (assignment) IoT networking standards (example topics): - LoRa: protocol specifications, energy consumption, modulation format, network design Review of wireless tools (example topics): - Introduction to wireless packet sniffing and analysis using Wireshark (assignment) - Simple simulations of WiFi network with NS3 RFID networking (example topics): - Principles of backscatter - Protocol formats: EPC C1G2 - RFID hackathon (assignment) Cognitive radio (example topics): - Basics of spectrum management - White Space Databases - Theory of spectrum sensing	
Study Goals	At the end of the course students will be able to: (i) to understand how practical wireless systems work and get a deeper understanding of how the theoretical concepts of wireless communications apply to practice; (ii) employ their own analysis methodology to assess new wireless network systems (especially at the physical layer); (iii) understand rapid prototyping of new wireless systems (for instance, with software defined radio).	
Education Method	Lecture presentations, mini-project assignments, assigned paper reading and its critical analysis and presentation.	
Computer Use	Each student should have its own laptop (preferably with a Linux distribution, where Linux must not be installed on a virtual machine). We will be using Matlab, and/or NS3 and/or GNURadio and/or Wireshark for the assignments.	
Books	WiFi Matlab WLAN toolbox: https://nl.mathworks.com/help/wlan/; Wireshark learn page: https://www.wireshark.org/#learnWS; tutorial on NS3 network simulator: https://www.nsnam.org/documentation/; specific chapters from books provided at the beginning of each lecture.	
Prerequisites	Background in programming (Matlab, Python, Bash)	
Assessment	Points from the mini-project assignments. A research paper analysis from conferences such as IEEE INFOCOM, ACM MobiCom, ACM SIGCOMM will be required to pass the course. disclaimer: information may change depending on the developments around the coronavirus.	

IN4152	3D Computer Graphics and Animation	5
Responsible Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. R. Marroquim	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students that haven't followed any previous Computer Graphics courses (like TI1806) will be able to participate, but might have to invest some more time to catch up in the first lectures.	
Course Contents	Have you ever wondered how Toy Story was made, why the game Last of Us 2 looks so beautiful, or have you ever wanted to create your own graphics application or game? Then you should consider following this course. If not, then you should still follow it... maybe, you will become interested! In this course, you will get a good idea of Computer Graphics in general. The topic is of very high relevance for the industry and the research community and has numerous applications in different domains, such as scientific visualization, video games, simulators, special effects, animated movies and many more. Here, you will learn about basic algorithms, as well as modern techniques. We will address several topics: the principles of image synthesis, object representations, geometric and hierarchical transformations, graphics cards and the graphics pipeline, realistic rendering (including global illumination and effects, such as reflections), expressive rendering, physics simulations, rendering control (including previsualization systems used by professionals in the movie industry), and perceptual rendering, which relies on properties of the human visual system to enhance the quality of the images. Besides course sessions on the theory of Computer Graphics, some of the algorithms will also be reproduced in practice, and deepened during the final project.	
Study Goals	The course teaches computer graphics techniques on an advanced level. After the course the student is able to classify the different modeling, shading, and display techniques. The student can reproduce the basic mathematical and algorithmic notions associated with these concepts, can comment on the weak and strong points of these techniques, and can apply the core concepts within a graphics program in practice.	
Education Method	lectures, instructions, research papers, lab work	
Literature and Study Materials	Research Papers in domain of selected topics, lecture sheets, online sources, optional books (see below)	
Books	Fundamentals of Computer Graphics by Shirley et al. - CRC Press Real-time Rendering by Tomas Akenine-Möller, Eric Haines, Naty Hoffman - Peters, Wellesley Real-Time Shadows by Elmar Eisemann, Michael Schwarz, Ulf Assarsson, Michael Wimmer - Taylor & Francis Computer Graphics. Principles and Practice by James D. Foley, Andries VanDam, Steven K. Feiner - Addison Wesley	
Assessment	The course will be evaluated with two grades, a project grade, accounting for 60%, and a paper grade 40%. The project grade is the result of a project and its presentation that is building upon the assignments that are handed out (roughly) weekly during the duration of this course. The paper grade is the result of the presentation of a scientific paper and the development of an associated practical implementation. Details of both elements will be presented during the lecture. Both grades (project and paper) have to be at least a 5.0 and their weighted average should be 6.0 or higher after rounding (0.5 steps).	

IN4253ET	"Hacking Lab"-Applied Security Analysis	5
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Necessary background differs per student project, see first lecture or contact instructors for details	
Course Contents	The security of computer and telecommunication systems is becoming an increasing concern. In this course, we will review the current state of the art on security research and gain practical experience in assessing the security and vulnerabilities of communication systems. Engineers are typically taught to focus on performance, correctness, scalability, and maintainability when building communication and information processing systems. However, an additional set of design principles are required to achieve security. In this course, we discuss security principles, common pitfalls and vulnerabilities. The weekly lectures provide an introduction into security research, with a focus on real-world security, privacy-enhancing technology and common security pitfalls. Each student participates in a "Hack Project", with a group of one to four students. Students can select between a wide range of available Hack Project outlines within the first week. The goal may be to evaluate the security of a real-world IT system, developing a proof-of-concept exposing a vulnerability or focussed on preserving privacy in a post-Snowden world. Students may propose their own Hack Project based on their background knowledge and skills. Such Hack Projects need to be approved and shaped together with the instructor. Example of possible outlined hardware-oriented projects are: development of a wifi tracker, programing an FPGA system to break passwords, assess the security of RFID cards, or to transparently intercept Ethernet traffic. Concrete software projects are: hacking Bitcoin, improving the TOR anonymity protocol and create Android-based tools for human rights activists in Iran, Egypt and Russia, reprogramming neural networks attacks. Each Hack Project is documented with a written report. This can be in the form of a 6-8 page IEEE-style scientific article or a traditional more lengthy report. All results, experiences and findings are presented to the entire class in the last week of the course. Hack Projects also report their progress several times during the course, after the weekly lectures.	
Study Goals	After this course, the student will have a thorough knowledge of security in real-world systems, and will be able to explore the literature on this topic independently. The student will be aware of the poor state of security in real-world computer systems. The student can explain the common pitfalls, why these known failures still occur and reasons behind the poor state of security in general.	
Education Method	Lectures, student presentations, written final report and active participation. Attendance and active participation during lectures is mandatory. This sadly means telelecturing is not possible.	
Literature and Study Materials	Customize literature lists and study materials are provided per project topic	
Assessment	The final class grade is composed of several partial grades. Partial grades are given for the written Hack Project report (60%), final presentation of result (10%), presentation of ongoing project progress (20%), participation in discussions, overall quality of the practical work and class attendance (10%). Students are required to obtain a passing grade on all partial grades. Attendance to lectures is mandatory. No final written exam. No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	If there is an unexpected high demand for this course, then enrollment will be based on past performance in relevant courses.	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4325	Information Retrieval	5
Responsible Instructor	Dr. C. Hauff	
Responsible Instructor	J. Yang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.	
Course Contents	Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs. Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines. Covered topics include: = Information Retrieval Models; = Indexing Techniques; = Web Search; = Information Seeking Paradigms; = Evaluation of information retrieval systems; = Components of Natural Language Processing; = Word Embedding Techniques; = Neural Language Models; = Deep Contextual Embeddings; = Neural Information Retrieval	
Study Goals	At the completion of this course, students will be able to: = Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1] = Describe and implement different indexing techniques. [Learning Objective 2] = Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3] = Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4] = Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5] = Describe typical NLP tasks and components of an NLP system. [Learning Objective 6] = Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7] = Compare word embedding and contextual embedding techniques. [Learning Objective 8] = Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]	
Education Method	Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments. Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.	
Literature and Study Materials	Scientific papers, course slides, course books - all resources are available on Brightspace.	
Books	Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA. Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press. Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson. Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing. Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".	
Assessment	Weekly individual assignment, weighting 10% of the final grade. Two group projects for IR and NLP, each weighting 45% of the final grade. The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities. disclaimer: information may change depending on the developments around the coronavirus.	

IN4343	Real-time Systems	5
Responsible Instructor	Dr. G. Iosifidis	
Instructor	Prof.dr. K.G. Langendoen	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures & 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	3TU MSc Embedded Systems; the corresponding courses are 2IN26 at TU Eindhoven, and 312030 at TU Twente	
Expected prior knowledge	Basic software engineering, C system programming, basic Linux operating system knowledge	
Course Contents	- basic concepts of RTS - worst case execution time estimation - scheduling policies - response-time analysis - jitter analysis - handling overload - multiprocessor scheduling - reservation-based scheduling	
Study Goals	The course intends to bring the student into the position to: - Explain the fundamental concepts and terminology of real-time systems - Construct task schedules using different scheduling policies under a given set of realistic system constraints - Analyze the timing behavior of a system for a given system model and scheduling policy - Discuss advantages and disadvantages of different scheduling policies for a given platform or system - Discuss the effect of hardware and software interferences on the timing behavior of a given system - Identify (reverse engineer) parameters of a scheduling scheme or a task set from output traces of the system - Derive (reverse engineer) the system specification from a given implementation (in the lab) - Evaluate the scheduling overheads of a given implementation (in the lab) - Implement event-based scheduling policies on a given microcontroller (in the lab)	
Education Method	lectures with exercises (32 hrs); self study (78 hrs); lab assignments (30 hrs)	
Books	Hard Real-Time Computing Systems by G.C. Buttazzo, Springer 2011	
Assessment	Written exam (grade) + lab work; the exam has a resit disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	3	
Permitted Materials during Tests	Simple calculator	

IN4391	Distributed Systems	5
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Starting with the mid-1990s, computing is undergoing a revolution, in which collections of independent computers appear to users as a single, albeit distributed, computing system. Motivated by the increase in the computation capacity of consumer computers, by the commoditization of server-grade machines, and by the advent of the Internet, the distributed computing paradigm has permeated all fields using computers. Current distributed computing applications range from the consumer social networks, peer-to-peer file-sharing, and massively multiplayer online games; to scientific computing using Big Data and distributed sensors; and to engineering fields and industrial control systems. This course focuses on the systems aspects of distributed computing. Specific, contemporary distributed systems are used as illustrative examples to discuss system design and non-functional requirements.	
Study Goals	1. Explain the objectives and functions of distributed computing systems. 2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems. 3. Describe the architecture and operation of distributed computing systems. 4. Explain how distributed computing systems can process user workloads. 5. Explain how distributed computing systems can detect and correct faults and errors. 6. Implement complex operations of modern distributed computing systems in realistic scenarios. 7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)	
Education Method	Lectures: This class uses a partially flipped classroom setting. Students are required to prepare parts of the material through self-study to be able to follow the in-class discussion. This involves, e.g., reading scientific papers. Practical: Designing, implementing, and evaluating a complete distributed system in groups, based on existing research work. Multiple topics are given to choose from. Deliverables include the code and a report of max. 10 pages.	
Literature and Study Materials	Textbook: Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Principles and Paradigms (2nd Edition), Prentice Hall, 2006. The textbook introduces the student to the traditional theory of distributed systems. Additional material: Several relevant research articles introduce the student to the classic literature as well as the latest advances on the topic.	
Assessment	Written exam (closed-book, open questions) during the exam period. A resit for the exam is offered in the following exam period. Practical project assessed based on the code, a presentation, and the report. This course uses gamification. Points can be collected through the practical project (max 4000 points) and the final exam (max 6000 points). The final grade is determined proportional to the 10000 points total. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 4 2021

CS4035	Cyber Data Analytics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	The course provides theoretical and practical background for applying data analytics in the field of cyber security. Cyber data analytics is a huge field with a great diversity of techniques and applications. The course is centered on a selection of seven such techniques: learning from imbalanced data; behavioral profiling and anomaly detection; sequential data mining; data stream mining; learning from software data; adversarial machine learning; and privacy-aware data mining; Anomaly detection is one of the main topics in cyber security. Specific difficulties that the student will learn to handle are the huge amounts of data and the large number of false positives. Behavioral profiling applies to both people and software processes. Different techniques will be taught to construct profiles from software logs. While building such profiles, care should be taken to not infringe upon the privacy of individuals the data is collected from. Finally, attackers will modify their behavior in order to avoid being detected, a cyber data analytics engineer tries to make their models/profiles robust against such modifications.	
Study Goals	The student will be able to: Apply machine learning to real data Understand and modify machine learning algorithms Learn models from time series Detect anomalies in multidimensional time-series Use distributed processing to speed up machine learning Learn models from data streams with limited memory Learn sequential models Use machine learning for fingerprinting and profiling Preserve the privacy of data owners while learning models Learn robust models that can detect evasive attackers Use machine learning to detect fraud, attacks, and botnets	
Education Method	There will be two lectures for each of the seven topics, and 3 large lab assignments on fraud detection, anomaly detection, and behavioral profiling, and 1 smaller lab on adversarial robustness. There is no exam. Teams of two students will work on these assignments which contain both individual and collaborative components. Deadlines are strict as peer-review will be used to both learn of other possible solutions, provide feedback, and get initial estimates on the obtained grade.	
Assessment	3 large lab assignments on fraud detection (30%), anomaly detection (30%), and behavioral profiling (30%), and 1 smaller lab on adversarial robustness (10%).	

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4125	Seminar Research Methodology for Data Science	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Instructor	Dr. K.A. Hildebrandt	
Instructor	J. Urbano Merino	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	basic knowledge in mathematics (linear algebra, calculus, probability and statistics)	
Course Contents	The course focuses on research methods for data science. It looks at underlying principles and concepts for data collection, analysis and data processing, as well as the use of tools to do this. The main topics of study are: Conceptualizing research questions and experimental design Frequentist and Bayesian data analysis Generalized linear models for statistical analysis Multilevel modelling for hierarchical and longitudinal data analysis Measuring and sampling, validity and reliability Linear and nonlinear dimensional reduction Principles of statistical testing In the course, students will be using software tools such as R, and Matlab/Mathematica	
Study Goals	The main aims of this module for the student is to achieve understanding of research methods for data science and obtain practical experience with data analysis and data processing methods. This module provides students with the opportunity to develop and demonstrate their understanding, knowledge, and competence. The learning outcomes for the module are that students will be able to: 1. Appreciate and comprehend strategies for collecting and processing data to answer data-driven research questions 2. Understand and reproduce key principles underlying statistical data and data processing analysis 3. Learn to identify and avoid typical biases, paradoxes and misunderstandings in data-driven research 4. Apply and select appropriate data modelling techniques to analyse data and data processing	
Education Method	Lectures/Assignments Expected Workload Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours (5 × 5 hours for each tool) Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours) Total = 140 hours	
Literature and Study Materials	Will be provided online	
Assessment	Course will be assessed on 3 coursework assignments. A) Analysis of experimental research data (40%) B) Exploration of real-world data set (20%) C) Linear and nonlinear dimensional reduction (40%) Students work in small groups on the 3 assignments. For each assignment, the student group submit a report and give a presentation including a question and answer round where individual group members are assessed on the coursework. The final course mark is the weighted average of the three assignment marks. Note that, there is a minimum grade of 5.0 for each assignment grade and an average grade for all components of at least a 5.8 in order to pass the course. Also, marks for individual assignments do not carry to the next year. Resit next quarter Resubmission of modified coursework is only allowed for assignments that received a fail mark (<5.0). Overall resit mark will be capped to 6.0. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	NA	

CS4140ES	Embedded Systems Laboratory	5
Responsible Instructor	Prof.dr. K.G. Langendoen	
Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/4 Lectures + 0/0/0/4 Lab	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	MUST have C programming skills. Students who have taken the CSE2425 Emb. Software course automatically qualify, others will have to pass an on-line ACCEPTANCE test.	
Course Contents	This highly multi-disciplinary course comes with a lab project where teams of 4 students each will have to develop an embedded control unit for a tethered electrical model quad rotor aerial vehicle (the Quadrupel drone), in order to provide stabilization such that it can hover and (ideally!) fly, with only limited user control (one joystick). The control algorithm (which is given) must be mapped onto a home-brew PCB holding a modern RF SoC interfacing a sensor module and the motor controllers. The students will be exposed to simple physics, signal processing, sensors (gyros, accelerometers), actuators (motors, servos), basic control principles, and, of course, embedded software (C) which is the programming language to be used in order to develop the control system. The project work (including written report) covers the entire duration of the course period, and will take approximately 128 hours, of which 32 hours are spent at the lab facilities. This is a core course of the Masters in Embedded Systems.	
Study Goals	Student is acquainted with real-time programming in an embedded context, along with a basic understanding of embedded systems, real-time communication, sensor data processing, actuator control, control theory, and simulation. Moreover, the student has had exposure to integrating the various multidisciplinary aspects at the system level.	
Education Method	Lectures (8*2hrs), lab work (8*4hrs), coding@home (8*12hrs), report (8hrs), so on average 2 days per week	
Literature and Study Materials	Lecture notes + Website	
Assessment	Lab. project (75%) + written report (25%), no exam, no resit disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	The capacity is limited and -as this is a compulsory course for ES students- they get preference over other MSc students.	

CS4145	Crowd Computing	5
Responsible Instructor	Prof.dr.ir. A. Bozzon	
Responsible Instructor	U.K. Gadiraju	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Basic knowledge of artificial intelligence and/or human computer interaction is advised. Proficiency in at least one programming language.	
Course Contents	Crowd Computing is an emerging field that sits at the intersection of computer science and data science. Crowd computing studies how large groups of people can solve complex tasks that are currently beyond the capabilities of artificial intelligence algorithms, and that cannot be solved by a single person alone. It involves the algorithmic engagement and coordination of people by means of Web-enabled platforms. These complex tasks are mainly focused on the creation, enrichment, and interpretation of data, making crowd computing a building block of data science. Examples of such tasks include the coordinated creation of data about real world events when electronic sensors are not available; the annotation of existing data sets to create ground truth data for the training of machine learning algorithms; and the analysis and interpretation of Web data to spot identify inappropriate content (e.g.,hate speech, or fake news). Crowd computing is an essential tool for any data-driven company: from Facebook to Microsoft, from Google to IBM, from Spotify to Pandora, all major companies employ crowd computing to fulfil their data needs, both by involving employees, and by reaching out to anonymous crowds through online marketplaces like Amazon Mechanical Turk or Appen. Crowd computing methods therefore play an important role in the design, development and evaluation of a variety of products, services, and systems in a variety of domains. The objective of the Crowd Computing course is to introduce the scientific and technical underpinnings of crowd computing, and to investigate how it can be used for computer science applications (e.g., information retrieval, machine learning, next-generation interfaces, and data mining) and for real world applications (e.g., cultural heritage preservation, online knowledge creation, smart cities, etc.) The course is designed around one key challenge, the creation and consumption of (high quality) data, and will be organized around three themes: 1) Establishing data needs; 2) Fulfilling data needs with crowd computing; and 3) Evaluating the quality of the retrieved data with respect to the original data need. Covered topics include: 1) Establishing Data Needs: - Requirement Elicitation - Requirement Analysis - User Modelling Properties 2) Fulfilling Data Needs with Crowd Computation: - Systems for/with collective intelligence (e.g., recommendation, semiautonomous systems, citizen science, crowdsourcing, and human computation systems) - Multi-modal Interaction (e.g., conversational systems) - Human Computation (e.g., worker modelling, task modelling, incentives, task assignment, recruitment) - Games with a purpose - Algorithms for Crowd Computing - Computational Methods for User Modelling - Interfaces for Crowd Computing Systems 3) Evaluating Retrieved Data: - Expert Evaluation - User Evaluation - Explanation of the output of Crowd Computing Systems 4) Study of Application Domains When applicable, the course will also feature invited lectures from selected academics and professionals in the field. Since instructors of this course are also directing the Design@Scale Delft AI lab, students of this course will have the opportunity to engage with cutting-edge research projects relevant to this lab. This Crowd Computing course is an elective for students following the Data Science and Technology Track and the Software Technology Track. It adds to the master education offer by addressing topics that are complementary to courses like IN4325 Information Retrieval, IN4252 Web Science & Engineering, CS4065 Multimedia Search and Recommendation, and IN4010 Artificial Intelligence Techniques.	
Study Goals	After this course, students will be able to: - Identify the requirements for a Crowd Computing system [LO1] - Design and develop Crowd Computing systems. Support and defend the relevance and correctness of his/her choices [LO2] - Describe and compare several Crowd Computing techniques. [LO3] - Describe and compare design decisions in the context of Crowd Computing interaction paradigms [LO4] - Determine which Crowd Computing technique(s) is most appropriate for being used in a certain problem domain [LO5] - Apply the appropriate Crowd Computing technique to an application domain and evaluate the obtained results. [LO6] - Analyse the performance of a Crowd Computing system by applying the proper evaluation measures. [LO7]	
Education Method	** NB: study guide information may change depending on the developments around the coronavirus. This course consists of 16 2-hour lectures. Each week, a 30-minute assignment tests the knowledge acquired on the discussed topics. Starting from Week 1, students form groups and work on a project, to be presented in week 9. Students are expected to work 6 hours per week (each) on the project assignment. Expected workload is 32 hours for attending lectures, 24 hours of reading study material and preparing lectures, 55 hours for weekly assignments and group assignment, 24 hours for preparing final survey, and 5 hours for exam and plenary presentations (total 140 hours).	
Literature and Study	Books:	

Materials	- Human Computation. Author(s): Edith Law and Luis von Ahn. Synthesis Lectures on Artificial Intelligence and Machine Learning, June 2011, Vol. 5, No. 3. http://www.morganclaypool.com/doi/abs/10.2200/S00371ED1V01Y201107AIM013 - A. Marcus and A. Parameswaran. Crowdsourced Data Management: Industry and Academic Perspectives. Foundations and TrendsR in Databases, vol. 6, no. 1-2, pp. 1161, 2013. DOI: 10.1561/1900000044. https://people.eecs.berkeley.edu/~adityagp/papers/crowd-book.pdf - An Introduction to Hybrid Human-Machine Information Systems. Demartini, G., Difallah, D.E., Gadiraju, U. and Catasta, M., 2017. Foundations and Trends in Web Science, 7(1), pp.1-87. https://edu.nl/np4th Slides: available on Brightspace Articles: available on Brightspace Recommended reading: - Interaction Design: Beyond Human-Computer Interaction (4th Ed, 2015). Authors: Jenny Preece, Helen Sharp, Yvonne Rogers
Assessment	The final grade consists of the following parts: - Weekly Individual assignment, weighting 15% of the final grade - Group assignment, weighting 55% of the final grade - Final Individual Assignment (Survey), weighting 30% of the final grade The group assignment is performed collectively, but graded individually. Assignments have no re-sit opportunities. Disclaimer: information may change depending on the developments around the coronavirus.
Tags	Algorithmics Artificial intelligence Design Programming Software
Co-Instructor	J. Yang

CS4205	Evolutionary Algorithms	5
Responsible Instructor	Prof.dr. P.A.N. Bosman	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Must have the ability to program your own (extensions of) evolutionary algorithms (e.g., C, C++, Java). Further, although no particular high-level prior knowledge is expected, a basic understanding of key aspects in linear algebra (e.g., matrix multiplications, inversions, decompositions, etc.), probability theory and statistics (e.g., normal distribution, statistical hypothesis testing), algorithm design (e.g., efficient local search, heuristics), complexity theory (e.g., big-O notation, algorithm analysis), will be very helpful.	
Course Contents	In this course we consider a specific subfield of Artificial Intelligence: Evolutionary Algorithms (EAs). These algorithms, sometimes also identified as being part of the class of bio-inspired algorithms, have as a metaphor the concept of natural evolution, i.e., the mechanisms by which, the fittest individuals in a population survive, reproduce, and in doing so, over time, change to be better equipped to thrive in their environment. Initiated in the 60s and 70s of the 20th century, research on EAs has progressed immensely. Today, EAs are being used to solve real-world problems in many areas, e.g. to optimize the layout of electrical wind farms, to automatically create radiation therapy treatment plans, and to optimize the architectures of deep neural networks. This course covers a spectrum of topics in EAs, ranging from basic concepts to advanced, recent, and state-of-the-art research, and ranging from theoretical to applied. In particular, topics include genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, optimal mixing evolutionary algorithms, multi-objective optimization, and real-world applications. The course is planned to have 7 lectures and 2 practical assignments. The first practical assignment pertains to experimenting with already implemented EAs on predefined problems. The second practical assignment offers more freedom, allowing you, in a group, to build your own EA (this may vary depending on student numbers and other circumstances). disclaimer: information may change depending on the developments around the coronavirus.	
Study Goals	Upon successful completion of this course, students will be able to: 1) Explain the key concepts underlying the main streams in Evolutionary Algorithm (EA) research, with in particular genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, and optimal mixing evolutionary algorithms. 2) Explain key ingredients underlying the rationale of when these algorithms work and when they do not work. In particular: schema analysis and how the match between the search bias of an EA and the fitness landscape is influenced by aspects such as variable dependencies and multi-modality. 3) Name and explain key research lines along which state-of-the-art research in EAs is done to achieve more robust, efficient, and effective EAs. 4) Identify good opportunities for using EAs, or hybrid versions thereof, in practice. 5) Properly (scientifically) experiment with EAs as well as program your own.	
Education Method	7 Lectures 2 Lab projects disclaimer: information may change depending on the developments around the coronavirus.	
Literature and Study Materials	Papers and slides that will be made available.	
Assessment	The final grade is based on 60% written exam, 40% lab practical work. disclaimer: information may change depending on the developments around the coronavirus. In particular, there may be no written exam. In that case, there will likely be several smaller practical assignments and a large practical assignment at the end of the course. Both parts will then likely count for 50%. For the large assignment there will also be chance to resit this part by means of a repair option through an oral examination of the lecture contents.	
Permitted Materials during Tests	None	
Tags	Algorithmics Artificial intelligence Optimalisation	

CS4210-B	Intelligent Decision Making Project	5
Responsible Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Dr. J.W. Böhmer	
Instructor	Dr. E. Demirovi	
Instructor	Dr. N. Yorke-Smith	
Instructor	Prof.dr. M.M. de Weerdt	
Contact Hours / Week x/x/x/x	0/0/0/1	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Theoretical knowledge regarding algorithms for decision making in Artificial Intelligence, obtained for instance by passing one of the following courses: - CS4210-A Algorithms for Intelligent Decision Making - CS4400 Deep Reinforcement Learning - IN4010(-12) Artificial Intelligence Techniques - IN4344 Advanced Algorithms.	
Course Contents	Decision making is at the centre of artificial intelligence. Building upon theoretical knowledge gained in other courses, students collaborate in small groups on a distinct research project per group, for instance on decision-making problems in transport, logistics or smart energy grids. Purely algorithmic challenges will also be provided. The research projects provide a good opportunity to learn about topics suitable for Masters projects in the Algorithmics section.	
Study Goals	After completing the Intelligent Decision Making Project course, the student is able to: 1. Apply algorithms for decision making to problem domains, and can compare and evaluate them. 2. Design and implement an extension of a decision-making algorithm. 3. Identify and discuss relevant topics in the research field of algorithms for intelligent decision making. 4. Describe and apply the appropriate research methodology. 5. Communicate his/her findings effectively.	
Education Method	A research project in a small group.	
Literature and Study Materials	Mainly survey papers and book chapters. Details are provided via Brightspace.	
Assessment	The assessment consists of the following items: 1. Quality of work of the research project (40%) 2. A scientific report of the research project (including peer review of a report) (20%) 3. Performance during the project (30%) 4. Oral presentation of the research project (10%) Only items 1 and 2 can be examined a second time. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Only a limited number of students can participate in this course. In order to be admitted, please submit a short motivation letter (max 200 words) via Brightspace. Attending the first lecture is compulsory.	
Tags	Artificial intelligence	
maximum aantal deelnemers	40	

CS4265	Computer and Network Security: Advanced Topics	5
Responsible Instructor	Prof.dr. M. Conti	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	*DISCLAIMER: study guide information may change depending on the developments around the coronavirus.* The course takes the form of seminars based on a selection of scientific papers (that either have had a strong impact on security today, or explore novel ideas that may be important in the future). The list of topics can be found in the brightspace Topics and Papers module. For each topic there is a primary paper, and possibly other additional papers. All the students are required to read all primary papers and be able to competently discuss the material in class. Each student will be responsible for presenting one lecture (based on one of the primary paper including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion in the class. 48 hours before each lecture each student must upload on a shared repository at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). This is intended to be an interactive class: class participation is strongly recommended (and will play a role in the grading criteria). Sleeping during the class is optional, but not recommended.	
Study Goals	This course is about learning to study, analize, do and criticize research in cybersecurity. This will be done by being exposed to actual research topics and scientific papers and discussing things together.	
Education Method	Studying, presenting and discussing recent research results in Computer and Network Security.	
Assessment	Presentation + Class Discussion + Written Report + Oral Exam (please refer to the Judgement field for more details) disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Judgement	The final grade will be made up of four components: 25% the presentation done by the student during the course: each student will be responsible for presenting one topic (based on the corresponding primary paper, including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion (Q&A) in the class. This component is based on following criteria: (15%) Layout and Graphics (30%) Content (20%) Organization (20%) Presentation (15%) Q&A. 25% for the active participation in Q&A sessions during the course: 48 hours before each lecture each student must submit (via email, to both the lecturer and the teaching assistants) at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). The students should actively participate in the discussion of the topics in the 10 minutes Q&A session for each presented topic. 25% for content and quality of the final essay: At the end of the course, each student must write a 5-page long essay about one of the topics that has been discussed in class, or another topic agreed with the lecturer. The topic and the structure of the essay must be agreed with the lecturer. The essay might include some implementation prototype or experiments/simulations to evaluate/support the claim in the paper (in case this is a significant part of the essay, two students can agree with the lecturer to work together). If the student cannot attend the lectures, an alternative work (e.g. a longer essay) must be agreed with the lecturer. 25% for the oral presentation of the essay: during the oral exam, the student is asked to give a 15-minute presentation to the lecturer and the teaching assistants about the essay (presenting with slides is highly recommended). During the oral presentation, students can also be asked questions about other topics of the course. This component is based on following criteria: (30%) Style (20%) Originality (50%) Organization (clarity in your argumentation, coherence between assumptions and conclusions, logical organization, evidence to support claims)	
Co-Instructor	Ir. S.E. Verwer	

CS4280	Language-Based Software Security	5
Responsible Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course has no formal prerequisites. However, for the homework assignments you will have to implement several program analysis techniques using the Scala programming language. If you have not used Scala before, you are thus expected to learn the basics of the language through self-study.	
Course Contents	Security vulnerabilities often arise due to programming errors in the source code of an application. Recent programming errors with severe security implications include Heartbleed (buffer over-read), Shellshock (code injection), and goto-fail (ill-formated code). Rather than hunt for individual vulnerabilities in programs, a more structural approach to improve security is to improve the programming language. This is the goal of language-based security: to rule out whole classes of potential security vulnerabilities in one go. This course studies various security properties and program analysis techniques for enforcing these properties at the level of the programming language to improve software security. In particular, we will study the following properties: - Memory safety: prevent buffer overflows and overreads - Type safety: prevent undefined behaviour - Information flow control: prevent data leaks and code injection attacks We will study techniques to address these problems at the language level through dynamic analysis, static analysis, and language design. To facilitate a precise study and comparison, we will define the above techniques formally in class. To facilitate student experimentation and exploration of trade-offs, students will implement the above techniques in homework assignments.	
Study Goals	After taking this course, students should be able to: 1. Describe the nature and causes of security vulnerabilities in software systems, and give concrete examples of how these security vulnerabilities can be exploited. 2. Explain the properties that can be enforced at the level of the programming language to rule out security vulnerabilities, such as memory safety, type safety, and non-interference. 3. Formally define the semantics of a simple programming language. 4. Formally define dynamic and static analysis techniques for enforcing these security properties. 5. Implement these techniques for a small programming language. 6. Discuss and evaluate the importance of soundness and precision of a given program analysis. 7. Contrast programming languages based on the set of countermeasures they provide, and give an appropriate recommendation for a specific application. 8. Analyse and apply results from scientific literature in the area of language based security.	
Education Method	The course work consists of the following activities: 1 or 2 instruction sessions per week. Weekly homework assignments consisting of theoretical questions, programming assignments, and reading assignments	
Assessment	The assessment for this course consists of two parts: The weekly homework assignments will test your ability to design an implement (variants of) the techniques discussed in the lectures (study goals 3-5). This counts for 40% of the total grade. The final written or oral exam will test your theoretical understanding of the security vulnerabilities and their countermeasures discussed in class (study goals 1-2) and your ability to discuss and contrast the different aspects of these techniques (study goals 6-8). This counts for 60% of the total grade. To pass the course, each of these grades (homework assignments and final exam) should be 5.0 or higher, and the final grade should be 5.8 or higher (and will be rounded to the nearest half grade point). disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	E. Visser	

CS4290	Seminar on Distributed Machine Learning Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	Machine learning systems are often conventionally designed for centralized processing in that they first collect data from distributed sources and then execute algorithms on a single server. Due to the limited scalability of processing large amount of data and the long latency delay, there is a strong demand for a paradigm shift to distributed or decentralized ML systems which execute ML algorithms on multiple and in some cases even geographically dispersed nodes. The aim of this seminar course is to let students learn how to design and build distributed ML systems via paper reading, presentation, discussion, and project prototyping. We provide a broad overview on the design of the state-of-the-art distributed ML systems, with a strong focus on the scalability, resource efficiency, data requirements, and robustness of the solutions. We will present an array of methodologies and techniques that can efficiently scale ML analysis to a large number of distributed nodes against all operation conditions, e.g., system failures and malicious attacks. The specific course topics are listed below. The course materials will be based on a mixture of classic and recently published papers. For each topic, the basic concepts and technology landscape will be first provided and then two state-of-the art of papers will be presented and discussed by students. We offer a testbed of a distributed (deep) ML system in which students are encouraged to apply different techniques to jointly improve its scalability and resilience. Course topics include Overview of distributed machine learning systems Performance and scalability of state-of-the-art systems Acceleration of ML workloads Slim distributed ML systems on small data Robust deep learning systems Federated machine learning systems	
Study Goals	Students are able to argue and reason about distributed ML from a systems perspective. Students understand the behavior and tradeoffs of distributed ML in terms of performance and scalability. Students can estimate the importance of data inputs via different techniques, i.e., core set and decomposition methods, for distributed ML systems. Students understand data poison attacks and design defense strategy for distributed ML systems. Students can analyze the state-of-the art federated machine learning systems and design the failure-resilient communication protocols. Students are able to design and implement methods and techniques for making distributed ML systems more efficient.	
Education Method	Lectures: 7 weeks X 2h Papers: one paper presentation, two paper reviews, and paper discussion. Practical: apply system and algorithmic optimization techniques learned in the lecture to improve the performance of distributed machine learning systems, e.g., image recognition on CIFAR 10. The testbed environment, learning algorithms, and dataset will be given. Deliverables include git commit of functioning code and a report summarizing the contribution	
Assessment	Paper presentation by group (10%): each group of 2 to 3 students needs to choose from a given set a papers to present (15 minutes) and lead the discussion (10 minutes). Paper reviews by individual (30%): each student needs to write three reviews of papers assigned from a given set and those two papers have to be different from the paper for the presented. Each review will account for 10% of the grade. Questionnaires by individual (0%): each student needs to hand in a list of questions at the beginning of the lectures that have paper presentation from other students. This is not graded. Individual project (60%): The project is collaborative among the entire class and competitive as a whole group. The objective is to continuously improve the performance of the given distributed ML system. The students need to hand in a final project report in style of a short scientific paper, stating their individual contribution to the overall system performance. disclaimer: information may change depending on the developments around the coronavirus.	

CS4295	Release Engineering for Machine Learning Applications	5
Responsible Instructor	L. Miranda da Cruz	
Responsible Instructor	Dr.ing. S. Proksch	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	The world of Software Engineering has been revolutionized in the last decade. Instead of releasing software updates yearly, companies can now release multiple times per week, sometimes even per day, to their customers. This allows much quicker reactions to market demands, software failures, and is crucial to increase the business value of software. These improvements have been mostly enabled by advances in release engineering and, in this course, we will learn about the techniques and technologies that build the foundation for modern release engineering. We will go on a journey that starts at continuous integration and then moves on to continuous delivery, continuous deployment, and continuous experimentation. We will discuss the theory and the current research on various related subjects like containerization, testing, or monitoring and will put the learned theory into practice. As a running example, we will build a pipeline for a machine learning application, which -compared to traditional release engineering- poses additional challenges, like data versioning or model deployment.	
Study Goals	After following this course, students are able to... - Apply standard techniques of release engineering - Apply version control techniques to machine learning artifacts, like data or models - Design a deployment pipeline for a machine learning application - Implement quality control techniques in a machine learning pipeline - Analyze and improve existing deployment pipelines - Evaluate and document design decisions in deployment pipelines	
Education Method	- Following interactive lectures - Active participation in tutorial sessions - Reading scientific papers and gray literature - Performing a small literature survey - Implementation of a pre-defined release engineering pipeline - Deriving and implementing an improvement for the pipeline - Documenting the improvement in a scientific essay	
Assessment	Formative Assessment: - Individual group meeting for feedback on current pipeline and pipeline extension proposal - Written feedback on Table of Contents and Introduction of written essay - Individual group meeting for feedback on project progress - Written feedback on methodology and pipeline of written essay Summative Assessment: 35% Final release engineering pipeline (focus: how well is the project executed) 60% Essay (focus: how well have design decisions been evaluated and documented) 5% Presentation (focus: clarification and fraud prevention) Please note: - The different parts of the summative assessment represent grading components and need ALL to be passed to receive a positive overall grade. - There is NO resit opportunity for this course. - Partial grades are not carried over to the next academic year.	
Special Information	The course information presented in the study guide may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. A.E. Zaidman	

CS4410	Category Theory for Programmers	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Categorical structures occur in programming languages on different levels: (1) within programming languages, providing design principles and guidance on how to write modular and correct-by-design programmes (as demonstrated in the practical programming language Haskell) and (2) in the design and study of programming languages, as a guiding meta-theory. In particular, category theory provides a mathematical justification for recursion schemes for inductive datatypes. This course aims to provide solid foundations on both (1) and (2).	
Study Goals	- Use categorical constructions (e.g., monads) in the design and structuring of computer programmes in Haskell - Prove properties of computer programmes, guided by categorical intuition - Understand categorical fusion laws and how to use them to optimize code - Understand the theory of infinite data structures and apply it to practical problems	
Education Method	Learning in this course is achieved through lectures, problem sessions, and guided self-study.	
Assessment	Exam at the end of the term, counts for 100% of the mark.	

EE4715	Array Processing	5
Responsible Instructor	Dr.ir. R.C. Hendriks	
Responsible Instructor	Prof.dr.ir. G.J.T. Leus	
Instructor	Prof.dr.ir. A.J. van der Veen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Linear algebra, signal processing, Fourier transform, stochastic processes and preferably statistical signal processing and some experience with matlab	
Summary	In this course we discuss array processing techniques for signal separation and parameter estimation, using arrays of sensors. After a review/introduction of the necessary linear algebra tools we will start with deriving the signal processing model for narrowband applications, followed by the wideband extension, and apply these to several applications among which array processing for wireless communication, audio and speech processing, biomedical signal processing and astronomy.	
Course Contents	Signal processing models for narrowband and wideband array processing, elementary beamforming concepts (spatial filtering), tools from linear algebra: QR, SVD, eigenvalue decompositions, projections and GEVD. Elementary beamformers/receivers: the matched filter, the Wiener filter, MVDR, LCMV, etc. Estimation of angles and delays using ESPRIT, adaptive space-time filters, the LMS algorithm and factor anaylsis.	
Study Goals	To be able to explain some key problems regarding data models, estimation and detection that occur in array processing applications. - To be able to explain the major signal processing tools required to solve array processing problems. - To be able to implement these signal processing techniques in Matlab. - To be able to apply these techniques to new array processing problems.	
Education Method	Lectures + mini project	
Literature and Study Materials	References from literature and notes	
Assessment	Oral exam: Take-home assignment with oral discussion of the results	

ET4030	Error Correcting Codes	4
Responsible Instructor	Dr.ir. J.H. Weber	
Contact Hours / Week x/x/x/x	0/0/0/3	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	A B.Sc. Programme in Electrical Engineering, Computer Science, or Mathematics	
Course Contents	Introduction into error-correcting codes; mathematical basics; block codes fundamentals; cyclic codes; co-operating codes; soft-decision decoding; convolutional codes; iterative decoding (turbo codes, LDPC codes); applications.	
Study Goals	The global goal of this course is to get acquainted with the basics and applications of error correction coding techniques. Such techniques are applied in order to protect information against errors which may occur during transmission or storage. The specific techniques under consideration in the course are the ones discussed in the lecture notes, which may be updated from year to year according to recent developments. The emphasis will be on the basic trade-offs between efficiency, reliability, and complexity. Unless explicitly indicated, the proofs of the results are not part of the course contents (the interested student may consult books from the bibliography). In the end, the student should be capable of making choices for suitable error correction coding techniques in the context of information transmission and storage applications. The student has to demonstrate to have understood the aforementioned techniques and trade-offs by solving exercises in a closed-book written or oral exam. The level of these exercises is similar to the examples and exercises provided in the lecture notes.	
Education Method	Lectures; expected workload is 22 hours attending lectures, 60 hours preparing for the lectures, studying the lecture notes, and making suggested exercises, and 30 hours for preparing and making the exam.	
Literature and Study Materials	Lecture notes "Error-Correcting Codes" by J.H. Weber	
Assessment	The final grade will be fully determined by a scheduled written exam, which will be held at the end of Q4. If it turns out that this is not possible, then an individual remote oral exam opportunity will be offered instead, also at the end of Q4. The resit will take place as a (remote) oral exam at the end of Q5, on appointment with the lecturer. The exam format is closed-book in any case.	
Remarks	The above-mentioned information may change depending on the developments around the Corona virus. Actual course information available on Brightspace.	

ET4285	Measuring and Simulating the Internet	4
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	(Advanced) Networking course (e.g., CS4055) and Programming skills.	
Course Contents	The Internet is a complex network without a fixed structure. Hence, measuring the Internet is crucial to acquire knowledge about the Internet infrastructure (topology), traffic, and performance (e.g., loss, delay, bandwidth, etc.). This course will discuss the design requirements and challenges in measuring and simulating the Internet, and the existing measurement methodologies (how/where/when to measure). Knowledge of how to conduct and evaluate Internet measurements enables the design and enhancement of a large set of applications, including: capacity planning and traffic engineering, network management and trouble-shooting, detecting network abuse and intrusions, etc.	
Study Goals	The goal of this course is to introduce the students to basic Internet measurement tools, as well as the state-of-the-art in Internet measurements research. The students will learn several Internet measurement techniques (e.g., active vs. passive measurements), and different software tools. Through a measurement assignment, the students will learn how to define/formulate a research problem, choose a specific approach, and complete a measurements-related research project.	
Education Method	Weekly instructions (8x2 hours) + independent project work (8x12 hours).	
Literature and Study Materials	Papers	
Assessment	Groups of students will be assigned a project that requires the students to put the theory on measuring and simulating the Internet into practice. The students have approximately 1 month to complete their assignment. The final assessment is based on the presentation (via report and/or demonstration) of the project assignment results and on the individual contribution and level of participation. Students within a group may thus receive different grades. As this is a project-based course, there is no official resit scheduled. Instead, an opportunity will be given to improve the work.	
Remarks	Disclaimer: The information about ET4285 may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	Because this is a project-based course, we can only admit a limited number of students (typically around 30, but the actual number depends on the number of TAs involved). If more students enrol, we will give preference to those who have successfully completed CS4055.	

IN4185	Globally Distributed Software Engineering	5
Responsible Instructor	Prof.dr.ir. D.M. van Solingen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	Exam by appointment	
Course Language	English	
Expected prior knowledge	Software Engineering (= IN2705)	
Course Contents	The course Globally Distributed Software Engineering (GDSE) will address pro's and con's of GDSE, practical consequences of GDSE, technological (in)feasabilities for GDSE, and practical experiences and examples of GDSE for example in outsourcing, off-shoring, near-shoring and multi-partner systems development. The central theme of this course is the fact that software engineering is carried out in practice more and more in globally distributed settings. This has advantages and disadvantages that need to be addressed in a practical matter when carrying such projects. The course is run asynchronous in BrightSpace. Lectures and excercises are followed digitally in weekly modules that need to be followed prior to the weekly synchronous lecture/virtual meeting. The course hours in the calendar are used for interaction with the professor and more detailed discussion and feedback. The course builds upon individual discipline in preparing the weekly modules online, in combination with group assignments during these weeks as well. Also the group assignments (in groups of 3 or 4 students) can be done virtually.	
Study Goals	The course Globally Distributed Software Engineering (GDSE) aims at teaching participants (1) the technical and organisational setting of carrying out software engineering in practice when distributed over the world, and (2) understanding best-practices in collaboration in software engineering project teams that carry out their work in a distributed setting.	
Education Method	Digital lectures, quizes, group assignments and online discussion. These are used as preparatory work prior to the weekly synchronous lectures (that are merely virtual as well), weekly group home work assignments and individual assignments.	
Computer Use	The course does not contain programming excercies. Though in the group assignment students will have to create a deliverable of choice. This can be very broad from creating a YouTube instruction video to writing an online book, or from creating a Wikipedia page to setting up tooling environment.	
Literature and Study Materials	Presentation handouts	
Assessment	Written report on lab work and literature research, individual f2f examination meeting of 30 minutes with professor. The course grade is calculated from the group assignment (25%), individual essay (25%), personal interview on GDSE course and individual essay (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Please enroll. If enrolled please pay attention that Module 1 of this course needs to be finished PRIOR to the first lecture meeting! Every week a new module is released in BrightSpace that needs to be worked through prior to the weekly synchronous meeting.	
Special Information	Please contact d.m.vansolingen@tudelft.nl	

IN4254	Smart Phone Sensing	5
Responsible Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Requirement 1: Students MUST either (1.1) have passed a JAVA programming course, or (1.2) have passed a C/C++ programming course and be familiar with JAVA, or (1.3) know Objective C (programming language for MACs). This requirement is equivalent to having passed the course TI 1206 in our first year Bachelor curriculum "Object Oriented Programming" Requirement 2: Students MUST (2.1) have passed a basic course on Probability Theory. This requirement is equivalent to having passed the course TI 2216M in our second year Bachelor curriculum "Probability and Statistics". We will be refreshing some concepts on Probability, but we will not be refreshing concepts on Object Oriented Programming.	
Course Contents	The course provides an introduction to the current research trends in the area of smartphones. The course will be based on a programming project, where students will form groups of two to develop a smartphone application. This is not a programming course; students are expected to have already programming experience. To develop a smartphone application, a user needs to be familiar with (1) the signals and data that smartphones can gather, and (2) the mathematical tools necessary to process this data. This course will provide a solid background for the above two points. During the lectures we will analyze the latest research papers on this emerging field. We will dissect these papers to understand how techniques from algorithms, signal processing and machine learning are used to develop some exciting applications. The students will then use these basic technical tools to develop their own apps.	
Study Goals	The goals of this course are twofold. First, to expose students to the increasingly important area of mobile computing. Students will learn how mobile phones can be used to solve problems in areas ranging from health care and indoor localization to song recognition and traffic management. Second, to provide students with a basic set of tools to develop their own applications. For students aiming for industry, the course should enhance their ability to use theoretical tools to solve practical problems. For students involved on research activities, the course will provide them with the necessary background to use smartphones as a distributed sensing and processing unit that could be used to solve the particular problems in their areas. After taking this course students will be able to: (1) Explain the current applications, methods and research trends in the area of smartphone sensing. (2) Apply key mathematical tools in the development of smartphone applications. (3) Analyze how a sensing and computing problem can be solved via the use of smartphones, and identify the steps required to design a solution. (4) Create a non--trivial and innovative smartphone application.	
Education Method	Lectures + Lab The project work, including the written report, covers the entire duration of the course period, and will take approximately 120 hours, of which 14 hours are spent on lectures, 10 hours preparing reports, 10 hours reading research papers, and the remaining part programming the App (the time spent in the Lab belong to this latter part).	
Literature and Study Materials	Research Papers and web tutorials	
Assessment	Written reports + project presentation + oral exam Overall, the final grade is determined by: 1) Two intermediate reports (5% of grade each, 2 pages each) 2) Final report (10 % of grade, 5 pages) 3) Final project demonstration (80% of grade) The first two reports are due on the third and fifth week; and the final report, project and exam are due on the ninth week. There is no resit for this course. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	1. You need to enrol in Brightspace 2. The first lecture will be compulsory 3. This course can only accommodate 60 students, with ES students having a preference when demand exceeds capacity. If your program marks this course as required, you are guaranteed a spot. IMPORTANT: The study guide information may change depending on the developments around the coronavirus.	

IN4255	Geometric Data Processing	5
Responsible Instructor	Dr. K.A. Hildebrandt	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Basic knowledge in mathematics (linear algebra, calculus): TI1106M, TI1206M or comparable courses. Students who haven't followed any of these courses can follow the course, but should be willing to invest more time.	
Course Contents	Geometry processing is concerned with the representation, analysis, manipulation, and optimization of digital shapes. Thanks to the advances in 3D acquisition and manufacturing technologies (like 3D-Scanning and 3D-printing), the usage of geometric data is continuously increasing and an efficient processing of digital shapes plays an important role for a variety of applications in areas such as computer graphics, computer-aided design and engineering, medical imaging and surgery planning, architecture, and entertainment. In this course, we will study concepts and algorithms for creating, analyzing, editing and optimizing digital geometric shapes.	
Study Goals	After successfully completing this course, the student is able to: - describe the fundamental techniques used for representing, analyzing, processing and modeling digital 3D-shapes treated in the course and to explain the mathematical and algorithmic concepts associated with them - apply the learned mathematical concepts to solve basic geometric problems arising in geometric modeling applications - design algorithms that can solve simple geometric modeling tasks and evaluate the drawbacks, benefits and limitations of the proposed algorithms - implement the designed algorithms in a geometric modeling software framework	
Education Method	The course combines lectures, tutorials, practical project work, and homework assignments.	
Literature and Study Materials	References to textbooks and recent research and survey papers are given in the lectures.	
Assessment	The course will be assessed on two practical projects and two theoretical assignments. The course grade is a weighted average of the grades of the practical projects (60%) and the theoretical assignments (40%). Note that, there is a minimum grade of 5.0 for each assignment grade and the average grade for all components of at least a 5.8 in order to pass the course. Also, grades for individual assignments do not carry to the next year. Resubmission of modified coursework is only allowed for assignments that received a fail grade (<5.0). Overall resit grades will be capped to 6.0 disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4331	Web-scale Data Management	5
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Dr. A. Katsifodimos	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Bachelor level courses in database management systems and operating systems. A prior course in distributed systems or middleware would be helpful but is not required. Programming skills are important for the final assignment.	
Course Contents	This course addresses the challenges of Data Management at Web-scale. Especially, it covers the need for large-scale distributed data storage systems. The lecture therefore introduces step-by-step increasingly complex distributed storage systems, leading up to modern implementations of different NoSQL data storage systems. The challenges arising from such systems are presented and discussed, especially focusing on the CAP theorem and the resulting trade-offs with respect to data models, transactional power, query expressivity, and replication consistency. These discussions lead to different variants of NoSQL database systems, like Key-Value Stores, Document Stores, Wide-Columnar stores, and Graph Databases. The advantages, disadvantages, and general properties of these systems are discusses in more detail. There is special focus on distributed transactions and consistency guarantees of different data management systems and methods.	
Study Goals	At the end of this course the student can - assess the nature of a given storage problem, and can select a suitable technology for solving it - understands the different data models encountered in Web Data Management, and their impact on modelling and querying - understands the issues arising from distributing and replicating data, especially with respect to the CAP theorem -understands the trade-offs which can be chosen within the design space of the CAP theorem -categorize and explain modern NoSQL databases within the framework of the previously mentioned trade-offs	
Education Method	Lectures and assignments	
Literature and Study Materials	Course slides and Lecture Videos	
Books	Literature mentioned in the lecture, mostly research papers.	
Assessment	Depending on the year, there are group assignments with final presentation, written report, and oral interviews. Final grade will be the weighted average of the three partial grades (the weights will be announced during the lecture depending on the available topics and assignments in the current quarter). The interview covers questions from the topics of the lecture. The interview can be resit, the assignment & presentation cannot. disclaimer: information may change depending on the developments around the coronavirus.	

IN4333	Language Engineering Project	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/0/0/4 (lab)	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Compiler construction CS4200-A and CS4200-B.	
Course Contents	"Software systems are the engines of modern information society. Our ability to cope with the increasing complexity of software systems is limited by the programming languages we use to build them. Bridging the gap between domain concepts and the implementation of these concepts in a programming language is one of the core challenges of software engineering. Modern programming languages have considerably reduced this gap, but often still require low-level programmatic encodings of domain concepts. Or as Alan Perlis formulated it in one of his famous epigrams: 'A programming language is low level when its programs require attention to the irrelevant'. A fixed set of (Turing Complete) programming constructs is sufficient to express all possible computations, but at the expense of considerable encoding that obfuscates the concepts under consideration. Linguistic abstraction can be used as a tool to capture our emerging understanding of domains of computation." (Visser, SCP 2015) In the precursor compiler construction course (CS4200), students learn the basics of language engineering by building a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts. In the precursor compiler construction course (IN4303), students learn the basics of language engineering and build a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts.	
Study Goals	In this course students learn to apply language engineering principles and tools to a real (domain-specific) programming language. Explore the definition of all aspects of a programming language: syntax, name binding, type analysis, transformations, code generation.	
Education Method	This is a project course. Students deepen their language engineering skills and insights by building a complete language definition. Students work in teams of two on the definition of a (domain-specific) programming language using the Spoofax Language Workbench. Assistance and feedback is provided during weekly lab hours. The project should span the full life cycle of language implementation including a test suite, IDE, code generator, and distribution of the result as an Eclipse plugin.	
Literature and Study Materials	- Documentation of the design and implementation of a specific language - Papers about language definition techniques	
Assessment	The work is assessed based on a code review of the language definition, a written report about the project, and a presentation in the final project workshop. The course has no resit. disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is based on the following components: - git repository with language project (40%) - written report about the project (30%) - presentation (slides) (30%)	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Track Data Science & Technology 2021

Introduction 1

1. An Individual Exam Programma (IEP) in this track consists of
a. a common core,
b. courses offered by the faculty EEMCS,
c. a seminar offered by the programme CS or a Literature survey (IN4306)
d. free electives,
e. a thesis project (IN5000 Final project) worth 45 credits and
f. if required, homologation.

The IEP must be drawn up in agreement with the thesis coordinator of the research group in which the student wishes to carry out his or her thesis project. The thesis coordinator is a member of the scientific staff of that research group.

the seminar of the research group in which the thesis is performed or the Literature Study (IN4306) is part of said IEP,

Free elective courses
c. the number of credits spent on free electives in said IEP is no higher than 25 credits,
d. the number of credits spent on homologation in said IEP is no higher than 15 credits,
e. at least 40 credits of the courses in the IEP (notwithstanding the thesis project) should be computer Science courses. A list of these courses is published annually in the digital study guide.

Free electives - language course list:
Up to 3 credits may be spent on language courses. These may only be chosen if required. Placement tests showing the necessity to take one or more of these courses must be taken and submitted to the master coordinator.

WM1101TU English for academic purposes-3 3
WM1135TU English for academic purposes-4 3
WM1136TU Written English for technologists-1 3
WM1102TU Written English fortechnologists-2   3
WM1137TU Spoken English for technologists-1 2
WM1112TU Spoken English for technologists-2 2
WM1115TU Elementary 1 Dutch for foreigners 3
WM1116TU Elementary 2 Dutch for foreigners   3
WM11117TU Dutch intermediate 1   3

The free elective space may also be used for an extra project:
TUD4040 Joint Interdisciplinary Project (JIP) 15

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Common Core DST 2021	
Introduction 1	Common Core MSc CS - DST (at least 20 EC): Choose 4 out of 9.

CS4035	Cyber Data Analytics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	The course provides theoretical and practical background for applying data analytics in the field of cyber security. Cyber data analytics is a huge field with a great diversity of techniques and applications. The course is centered on a selection of seven such techniques: learning from imbalanced data; behavioral profiling and anomaly detection; sequential data mining; data stream mining; learning from software data; adversarial machine learning; and privacy-aware data mining; Anomaly detection is one of the main topics in cyber security. Specific difficulties that the student will learn to handle are the huge amounts of data and the large number of false positives. Behavioral profiling applies to both people and software processes. Different techniques will be taught to construct profiles from software logs. While building such profiles, care should be taken to not infringe upon the privacy of individuals the data is collected from. Finally, attackers will modify their behavior in order to avoid being detected, a cyber data analytics engineer tries to make their models/profiles robust against such modifications.	
Study Goals	The student will be able to: Apply machine learning to real data Understand and modify machine learning algorithms Learn models from time series Detect anomalies in multidimensional time-series Use distributed processing to speed up machine learning Learn models from data streams with limited memory Learn sequential models Use machine learning for fingerprinting and profiling Preserve the privacy of data owners while learning models Learn robust models that can detect evasive attackers Use machine learning to detect fraud, attacks, and botnets	
Education Method	There will be two lectures for each of the seven topics, and 3 large lab assignments on fraud detection, anomaly detection, and behavioral profiling, and 1 smaller lab on adversarial robustness. There is no exam. Teams of two students will work on these assignments which contain both individual and collaborative components. Deadlines are strict as peer-review will be used to both learn of other possible solutions, provide feedback, and get initial estimates on the obtained grade.	
Assessment	3 large lab assignments on fraud detection (30%), anomaly detection (30%), and behavioral profiling (30%), and 1 smaller lab on adversarial robustness (10%).	

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4220	Machine Learning 1	5
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

IN4089	Data Visualization	5
Responsible Instructor	T. Höllt	
Instructor	Prof.dr. E. Eisemann	
Contact Hours / Week x/x/x/x	0/2/0/0 & lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	Master course MKE/ST/DS	
Expected prior knowledge	IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required). The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.	
Course Contents	Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization, The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines. As a computer science course, affinity to algorithmic thinking and programing skills will be needed. Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.	
Study Goals	The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data. Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more. By the end of the course, you should be able to LO1: Discuss a large range of visualization techniques. LO2: Discuss a perception principle of visualization. LO3: Explain mathematical principles and algorithms of visualization techniques. LO4: Design suitable visualization systems for a given practical data analysis problem. LO5: Implement visualization systems for a given practical data analysis problem.	
Education Method	Lectures, practical assignments, self-study, and projects.	
Literature and Study Materials	Course slides, instructions for projects, and selected literature. Chapters from: Visualization Analysis and Design Author: Tamara Munzner CRC Press Visual Computing for Medicine 2nd Edition Theory, Algorithms, and Applications Authors: Bernhard Preim Charl Botha Morgan Kaufmann All available in electronic form via Brightspace or at TUDelft library.	
Assessment	The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Judgement	The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam. The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project. Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions. The project is evaluated based on the developed result, its documentation and presentation. Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.	

IN4252	Web Science & Engineering	5
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

IN4391	Distributed Systems	5
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Starting with the mid-1990s, computing is undergoing a revolution, in which collections of independent computers appear to users as a single, albeit distributed, computing system. Motivated by the increase in the computation capacity of consumer computers, by the commoditization of server-grade machines, and by the advent of the Internet, the distributed computing paradigm has permeated all fields using computers. Current distributed computing applications range from the consumer social networks, peer-to-peer file-sharing, and massively multiplayer online games; to scientific computing using Big Data and distributed sensors; and to engineering fields and industrial control systems. This course focuses on the systems aspects of distributed computing. Specific, contemporary distributed systems are used as illustrative examples to discuss system design and non-functional requirements.	
Study Goals	1. Explain the objectives and functions of distributed computing systems. 2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems. 3. Describe the architecture and operation of distributed computing systems. 4. Explain how distributed computing systems can process user workloads. 5. Explain how distributed computing systems can detect and correct faults and errors. 6. Implement complex operations of modern distributed computing systems in realistic scenarios. 7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)	
Education Method	Lectures: This class uses a partially flipped classroom setting. Students are required to prepare parts of the material through self-study to be able to follow the in-class discussion. This involves, e.g., reading scientific papers. Practical: Designing, implementing, and evaluating a complete distributed system in groups, based on existing research work. Multiple topics are given to choose from. Deliverables include the code and a report of max. 10 pages.	
Literature and Study Materials	Textbook: Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Principles and Paradigms (2nd Edition), Prentice Hall, 2006. The textbook introduces the student to the traditional theory of distributed systems. Additional material: Several relevant research articles introduce the student to the classic literature as well as the latest advances on the topic.	
Assessment	Written exam (closed-book, open questions) during the exam period. A resit for the exam is offered in the following exam period. Practical project assessed based on the code, a presentation, and the report. This course uses gamification. Points can be collected through the practical project (max 4000 points) and the final exam (max 6000 points). The final grade is determined proportional to the 10000 points total. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 1 2021

AP3421	Fundamentals of Quantum Information	4
Responsible Instructor	Dr. L. di DiCarlo	
Instructor	Dr. D. Elkouss Coronas	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Knowledge of linear algebra, probability and statistics.	
Course Contents	Approximate syllabus: - quantum states, unitary operations, and measurements; - universal gate sets; - entanglement, Bell test; - basic quantum communication protocols; - basic algorithms and quantum algorithmic techniques; - basic quantum error correction; - simple physical implementations of qubits.	
Study Goals	Motivation: Quantum information is the future of computing and communication. Quantum computers offer exponential speedup over any classical computer. Similarly, quantum communication offers many advantages, including the ability to create secure encryption keys where security rests only on the laws of nature. Synopsis: This class will teach you the fundamental principles of quantum information. You will learn essential concepts that distinguish quantum from classical devices. You will learn about quantum bits and the quantum operations and measurements that can be performed on them. You will learn the basic techniques used in quantum algorithms, and examine basic examples of such algorithms. You will also take the first step in understanding how a quantum bit can be physically implemented. Aim: To learn the fundamental concepts underlying quantum computation and communication systems.	
Education Method	3 hours of lecture, 1 hour tutorial per week.	
Literature and Study Materials	The main reference textbook for the course is Nielsen and Chuang, Quantum Computation and Information, Cambridge University Press.	
Assessment	30% homework assignments, 10% in class quiz, 60% final exam. A minimum grade of 5.0 (unrounded) for the final exam is required to pass the course.	
Permitted Materials during Tests		
Continuing Courses	This course forms part of the curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information AP3421-PR Quantum Information Project CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation	

CS4070	Multivariate Data Analysis	5
Responsible Instructor	Dr.ir. F.H. van der Meulen	
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	4/4/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	

Expected prior knowledge

* Introduction Probability Theory and Statistics: see for instance

A Modern Introduction to Probability and Statistics
Understanding Why and How
Series: Springer Texts in Statistics
Dekking, F.M., Kraaikamp, C., Lopuhaä, H.P., Meester, L.E.
2005, XVI, 488 p. 120 illus., Hardcover
ISBN: 1-85233-896-2

* Basic calculus

* Linear Algebra: matrix multiplication, the inverse of a matrix, the transpose of a matrix, least square solution.
see:

David C. Lay: Linear Algebra and Its Applications
ISBN-10: 0321385179 ISBN-13: 9780321385178
©2012 Pearson)

Course Contents

PART I:
Stochastic models will be developed on the basis of probability theory. Probability theory describes the behavior of certain phenomena in terms of how likely it is that certain values will occur. Central features of the models will be discussed are random variables, probability density functions, and the expected value operator. In describing random processes and signals, the correlation function and conditional probabilities play a central role.

It addresses the following subjects:
1. Random variables. Matlab exercise on estimation of PDF, expected value and variance.
2. Refresher correlation. Calculating with correlation functions.
3. Random processes, correlation function, stationarity, wide sense stationarity, estimation of correlation function (Matlab exercise).
4. Random signal processing, power spectral density function, white noise.
5. AR processes, linear prediction: theory and Matlab exercise.
6. Markov chains.

PART II:
A course in advanced statistics about linear models, Bayesian inference, classification problems, Gaussian processes and Markov Chain Monte Carlo.

Study Goals

PART I:
1. Probability Theory
- Conditional) probabilities, the law of total probability, and Bayes rule.
- Solve probability problems that require the use of axioms of probability.

2. Definition and Description of Random Variables and Processes
PDF, PMF, CDF, Covariance, Correlation- Determine if a given PDF, PMF, CDF, variance, (auto/cross-)correlation(-function), (auto/cross-)covariance(-function), power spectral density complies with (theoretical and analytical) requirements.
- Convert the description of a probabilistic problem into a probabilistic model using PDF, PMF, or CDF.

3. PDF/PMF and Expected Value
Calculate the various forms of expected value of (combinations of) random variables and random processes
- For a given (amplitude continuous/discrete and time continuous/discrete) probability model calculate the following probabilistic (marginal, joint and conditional) characterizations: PDF, PMF, CDF, probability of an event, expected value, variance, covariance, correlation, correlation coefficient, auto/crosscorrelation function, auto/crosscovariance function, (cross) power spectral density.
- Calculate the PDF, PMF, expected value and variance of a derived random variable.

4. Properties of Random Processes
- Independence, orthogonality, uncorrelated, whiteness, IID- Determine if random variables/processes have the following properties: independent, orthogonal, uncorrelated, white, Poisson, Gaussian, Bernoulli, Markov, IID, stationary, WSS, ergodic.
- Calculate the expected value, variance, auto/crosscorrelation(function), auto/crosscovariance(function), power spectral density of a linear combination of random variables and of a linearly filtered (WSS, amplitude discrete/continuous, time discrete/continuous) random process.

5. Large NumbersCentral limit theorem, law of large numbers
- Solve problems that require the use of the central limit theorem in an engineering context
- Explain the law of the large numbers in an engineering context.

6. Statistical Estimators
- Estimated mean, variance, and correlation function
- Given a set of outcomes, sample functions or realizations, calculate estimators for expected value, variance, and (auto-)correlation function.

12. Application to Engineering Problems and Simulations
- Select and translate a simple electrical engineering or computer science problem into mathematical probability model. The emphasis is on problems in signal and image processing, telecommunication, and media and knowledge technology. The class of probability models encompasses the following random variables/processes: Bernoulli, exponential, binomial, Poisson, Gaussian, uniform.
- Justify and reflect on the approach taken in calculating or simulating (MatLab) the following probabilistic properties: PDF, PMF, expected value, variance, autocorrelation function, autocovariance function.

	PART II: After finishing this course, the student is able to apply and derive statistical methods from both the frequentist and Bayesian perspective for - linear models - classification problems - clustering problems - Gaussian process regression The student is able to give a clear presentation about the underlying statistical theory. The student is able to compute several statistical characteristics by hand.
Education Method	PART I: Lectures, working groups (problem solving), laboratory work (a Matlab exercise) Workload is around 15 hours for attending lectures, 5 hours of reading study material and preparing lectures, 15 hours for the lab course, 20 hours for preparing the exam, 3 hours for the exam, and 8 hours for a final report (66 hours in total). PART II: Classes and weekly exercises.
Books	PART I: R.D. Yates and D.J. Goodman, "Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers", ISBN 0-471-17837-3, John Wiley and Sons, New York, 2005, Second Edition. PART II: Simon Rogers and Mark Girolami "A first coruse in machine learning, 2nd edition" Chapman & Hall From this book chapters 1--4, 8 and 9 will be covered.
Assessment	The final grade is the average of the grades you get for part (I) and (II). For part (I) there is a lab and written exam, where the grade is determined by the exam, and the lab assignment should be Passed. If you fail the lab assignment, you'll get a second chance to submit around the time the resit. For part (II), there will be an on-campus written exam. To pass the course, the average should be 5.8 or higher, and the grade of each individual part should be a 5.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.
Exam Hours	PART I: Online exam of 3 hours. PART 2: On campus 3 hour written exam PART II: Written exam of 3 hours.
Permitted Materials during Tests	PART I: Self made notes on a two-sided written A4 sheet. Calculator. PART II: none
Remarks	PART II: This course is particularly interesting for students that are interested in statistical exploratory and quantitative techniques to analyse multivariate data.

CS4200-A	Compiler Construction	5

Responsible Instructor
E. Visser

Contact Hours / Week x/x/x/x
6/0/0/0

Education Period
1

Start Education
1

Exam Period
1
2

Course Language
English

Expected prior knowledge
- programming (required)
- software engineering (recommended)
- concepts of programming languages (recommended)
- formal languages and automata (recommended)

Course Contents

Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation.

This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments.

The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler.

In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2).

Course CS4200-A covers the following topics:

* Syntax and parsing
  - concrete syntax, abstract syntax
  - context-free grammars
  - derivations, ambiguity, disambiguation, associativity, priority
  - parsing, parse trees, abstract syntax trees, terms
  - pretty-printing
  - parser generation
  - syntactic editor services

* Static semantics and type checking
  - name binding, name resolution, scope graphs
  - types, type checking, type inference, subtyping
  - unification, constraints
  - semantic editor services

Study Goals

After this course, students should be able to:

* Describe the architecture of a compiler and programming environment and the role of each component of that architecture

* Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages

* Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples

* Define the syntax of a programming language using declarative syntax definition that describes the concrete and abstract syntax of a programming language

* Define basic editor services

* Define the type system (name binding and typing rules) of a programming language using constraint generation rules

* Construct tests for each of the components of a compiler in order to determine its correctness

* Integrate the components into a working compiler and programming environment

Education Method

Attending the course involves
- attending weekly lectures
- attending weekly lab session (which may start with a group tutorial)
- reading lecture material and papers
- making homework assignments
- building the front-end of a compiler

Literature and Study Materials

Lecture slides and selected papers from the literature

We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments.

Reading material and homework assignments will be published on the course website; see
http://pl.ewi.tudelft.nl/education/compiler-construction/

We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.

Assessment

The final grade will be determined by the following components
- final exam (50%)
- course project (50%)

	Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	No materials are permitted during the exam.
Judgement	to be decided

CS4215	**Quantitative Performance Evaluation for Computing Systems**	**5**
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	Todays computing systems become ever complex, due to the rapid development of hardware and software technology. It is challenging to design and run computing systems that guarantee users performance requirements in a resource efficient way. Various quantitative methods are applied to capture such complex system dynamics and predict metrics of interests, from the designing phase of the systems to the runtime performance, e.g., job response times and system anomaly. To optimize the performance of computing systems, a deep understanding on those methods and their applications on the system design are essential. Having practical hand-on experience on designing experiments, deriving models, and validating results with benchmark systems will prepare students to tackle challenges of real world computing systems.	
Study Goals	LO1. Design full/fractional factorial experiments for multi-variate regression analysis, e.g., finding critical parameters for deep learning clusters LO2. Apply queueing theory to analyse and predict run-time performance of applications, e.g., the average response times of on-line ML training service LO3. Apply machine learning models to analyse and predict the system dependability, e.g,root cause analysis for machine failure. LO4. Conduct experiments to profile applications and extract their workload parameters on real systems, e.g., e.g., deep learning clusters LO5. Develop resource management policies and validate them on real computing systems, e.g., deep learning clusters	
Education Method	Lectures: 7 weeks X 2-4h Practical: Derive, validate and evaluate performance models and resource management strategies for a chosen system via homework and group project. Multiple types of computing and network systems can be chosen from. Deliverables include a report and group presentation.	
Books	1) Performance Modeling and Design of Computer Systems: Queuing Theory in Action by Mor. Harchol-Balter 2) Design and Analysis of Experiments by Douglas C. Montgomery, 3) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics.	
Assessment	Homework (40%): 2 individual homework Group project (60%): group project report and presentation disclaimer: information may change depending on the developments around the coronavirus.	

CS4270	Conversational Agents	5
Responsible Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Instructor	Dr. M. Bruijnes	
Instructor	Dr. P.K. Murukannaiah	
Instructor	M.L. Tielman	
Co-responsible for assignments	F. Broz	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic programming skills (e.g. Python and Java) Probability theory and statistics	
Course Contents	Chatbots, embodied and conversational virtual agents, and social robots are becoming more and more popular. Many people are owning an Alexa, Cortana or Echo or are talking to their virtual assistant on their phone. Indeed, such technologies have the potential of making our lives easier and relieve people from the more repetitive tasks. For example, it is imaginable that such systems are being used for financial applications by helping customers with frequently asked questions but also to advise them on in the long term more impactful decisions such as their pension plans. Further applications can be imagined in the area of healthcare and education, some of which are already in existence today. In this course, attention will be given to different verbal and nonverbal behavioral characteristics, like speech, intonation, gaze and gestures that humans show when communicating with both other people and machines. This behavior is then related to different dialogue functions, including turn-taking, addressing others, and backchanneling, that give shape to the communication process. This course introduces conversational agent technology. We cover agent related technologies which can be grouped into: Dialog Management NLP speech synthesis social robotics	
Study Goals	After this course you have learned to: 1) Apply relevant linguistic and psychological theory to conversational agent systems 2) Analyse human-human conversational data to better design ML models 3) Explain which components are part of a dialog system and what distinguishes rule-based from statistical dialog systems 4) Describe the design and implementation of state-of-the-art conversational agents, give examples of their application areas and analyse and discuss the limitations of current systems 5) Evaluate the effects of affect and embodiment on human-agent interaction 6) Create and evaluate a socially-aware conversational agent by applying state of the art tools and libraries	
Education Method	There are 2 lectures and 1 practicum scheduled per week. Students work in groups of 3-4 on a group project. Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours)	
Literature and Study Materials	We use the book "The conversational interface " by Michael McTear, Zoraida Callejas, David Griol. This book is freely available through the TU Delft library. https://link-springer-com.tudelft.idm.oclc.org/book/10.1007%2F978-3-319-32967-3 Other relevant material will be provided on Brightspace.	
Assessment	Online Examination (30%) Group Assignment (50%) (This assignment will result in a group report and a group online demonstration with individual question/answer part) Group presentation (20%) The exam and the assignments are both intermediate results, and will not count separately for the next academic year. Only the final grade will remain. A passing final grade for this course can only be earned when for the online examination and the group assignment at least a 5,0 is earned, and the average grade for both is at least a 5,8. Projectwork with a mark lower than 5.8 can be modified and resubmitted. The mark for resubmitted coursework will be capped to 5.8 Note that individual marks for projectwork or written exam do not carry to the next year. We further grade the labs as pass/fail. By a successful pass of all labs a bonus of 0.3 will be awarded towards the group assignment grade. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr.ir. W.P. Brinkman	

EE4C06	Networking	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	PART 1: Basics, concepts and computations of networks 1. Basics of networking & introduction to Network Science - what is a network? Representation of a graph, basics of graph theory, overview of the relatively new theory of complex networks, called Network Science. - important characterizers of a network (network metrics) - basic network/graph models - examples of real-world networks (airline transportation, the web and Internet, social networks, brain networks, etc.) and applications of network science 2. Concepts of networking - routing - Quality of Service (QoS) - traffic management and scheduling - network robustness (failure, cascading effects,...) - overlay networking and new aspects of networking such as interdependent networks PART 2: Applications and examples of networks (as listed below) will be taught (some of those by a guest lecturer). Ranging from year to year, a selection among the following will be covered: 1. Electrical networks (smart grids) 2. Networks on Chip (NoC) 3. Optical networks 4. Computer Networks (the Internet) 5. Mobile communication networks 6. Sensor networks 7. Biological networks 8. Social networks	
Study Goals	The course on Networking aims to provide a general and basic introduction to the art of networking, that tries to unravel the operation and behavior of networks, both man-made (infrastructures such as the Internet and power grids) as well as networks appearing in nature (such as the human brain, biological networks and social human interactions). The course on Networking will introduce concepts of the new Network Science, that basically studies the interplay between, on the one hand, the processes (also called functions or services) on the network and on the other hand, the underlying topology, that is mostly changing over time as an evolving organism, rather than as given or fixed object. Network Science combines many disciplines such as graph and network theory, probability theory, physical processes, control theory and algorithms. After this course, students are expected to represent/abstract real-world infrastructural network (e.g. a communication system) as a complex network, understand the basic methods to analyze properties of networks and dynamic processes on networks. Students will also understand why processes on networks and design of networks are so complex. Finally, students may appreciate the fascinatingly rich structure and behavior of networks and may realize that much in the theory of networks still lies open to be discovered.	
Education Method	Lectures, slides & homework	
Assessment	written examination disclaimer: information may change depending on the developments around the coronavirus.	

ET4388	Ad-hoc Networks	5
Responsible Instructor	Dr. R.R. Venkatesha Prasad	
Contact Hours / Week x/x/x/x	3/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Wireless communications and networking Computer communication principles, Layering principle of Computer Networks. Digital communication.	
Course Contents	IMPORTANT NOTICE ------------------------- Please note that the prevailing conditions of the COVID19 may force us to modify the study guide information. It may change depending on the developments around the coronavirus. We may have to make changes to the teaching methodology, assessment, practical work, assignment and group activities. This will be instructed by the faculty/university management based on the orders of the government from time to time to protect the faculty and students. The above applies to all the fields in this coursebase for this course. The course will be offered ONLINE. Face to Face meeting may be possible depending on the situation [Safety of everyone is the highest priority]. -- Ad-hoc networks are formed in situations where mobile computing devices require networking applications when a fixed network infrastructure is not available or not preferred to be used. In such cases, mobile devices may possibly set up an ad hoc network themselves. Ad-hoc networks are decentralized, self-organizing networks and are capable of forming a communication network without relying on any fixed infrastructure. Ad-hoc networks form a relatively new field of research. In this lecture, besides general introduction to ad-hoc networks and their applications, we will focus on state-of-the-art methods and technologies for forming an ad-hoc network and maintaining its stability despite the dynamics of the network. The contents of the course are as follows: Positioning and applications (Chapter 1, 2 & 3 of the textbook, these topics are basics & pre-requisites; And Chapter 5) o Definition of ad-hoc networks o Comparison with infrastructure based systems o Typical applications o Advantages and challenges o Radio technologies for ad-hoc networks o Wi-Fi, Zigbee, Bluetooth Modelling ad-hoc networks o Propagation models o Topology models based on graph theory o Degree and hopcount o Connectivity theorems MAC protocols for ad-hoc networks (Chapter 6, 10 of the textbook) o Introduction to MAC protocols o Issues and design goals o Classification o Directional, muli-channel MAC protocols o Energy efficiency in MAC protocols o Quality of service Self organisation and Routing (Chapter 7, 8, 11 of the textbook) o Flooding o Node discovery, neighbour discovery o Route establishment o Topology maintenance, localisation o Proactive, reactive and hybrid routing o Typical protocols o Energy efficiency in routing o Broadcast and multicast o Effects of mobility on connectivity and capacity o Effect of nodes joining and leaving the network Advanced issues in ad hoc networks o Wireless sensor networks (Chapter 12 of the textbook and papers) o Cooperation (Reference papers) o Simulating ad hoc networks as part of project (optional: ns3, OMNET, OPNET) o Energy Harvesting Project presentations by students	
Study Goals	By the end of this course students should be able to: - Model the ad-hoc networks using Graphs. - Describe the working principles of medium access control protocols for ad-hoc networks - Explain the working principles, advantages and disadvantages of different classes of routing protocols for ad-hoc networks - Choose various components to form a coherent ad hoc networking architecture - Develop a simulator to evaluate the MAC and routing protocols for ad hoc networks - Assess the suitability of ad-hoc networks for different communication needs and scenarios	
Education Method	The course will be taught in lecture form. The presence of students at all lectures is required for optimum result. Students are required to participate actively in various forms of activities and peer-learning. New forms of teaching aids are used.	
Literature and Study Materials	1. Textbook: Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004.	

	2. Lecture notes consisting of slides presented at the lectures (Slides are only teaching aid and they are not substitute for textbooks, research papers, etc). 3. Some recent journal papers 4. Optional Reference Books 4.1. Distributed Algorithms, Nancy A. Lynch, Morgan Kaufmann, 1996 (for networking algorithms) 4.2. Ad Hoc Mobile Wireless Networks, Principles, Protocols and Applications by Subir Kumar Sarkar , C Puttamadappa , and T. G Basavaraju, Auerbach Publications, 2008. This book is avaliable online in the library. 4.3. Wireless Ad Hoc and Sensor Networks, A Cross-Layer Design Perspective by Jurdak, Raja, Springer, 2007. This book is avaliable online in the library. 4.4. Ad-hoc Networks: Fundamental Properties and Network Topologies, by Ramin Hekmat, Springer. 5. OPNET/ns-2 web pages, tutorials and video lectures
Books	Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004. However, I also use other materials from Internet and other books listed above.
Assessment	1. There will be written tests/examinations for this course. 2. The students will carry out a project in a group and submit a short report. 3. Participation in off-track discussions on Facebook/Brightspace/FeedbackFruits and wikis. Final score is based on marks obtained during tests, project, assignment (in groups) and bonus marks. All the details will be given in the first class. Breakup: 2 Tests + Final Exam = 55% Project 40% Self assessment + Reflection 3% Activities on Feedback Fruits or any Online platform 2% ====================== Changes due to COVID19 in 2021 ================ There are three chapters Chapter 1: Network modelling Chapter 2: MAC protocol Chapter 3: Routing + Lab: Simple experiments using laptops (individual) + simulations (group-wise) Marks Breakup -------------- Homework/Assignments/Group works Part A1: Group-wise assignment. 3 times (1 per chapter) -- 15 marks Part A2: Group-wise Q&A. 3 times (1 per chapter) -- 30 marks Part B1: Individual experiment+report Part B2: Group-wise simulations + report + demo Part B1 + Part B2 - 50 Marks Part C: Self-assessment + peer activities -- 5 marks Resit: Part A1 & A2 will not be repeated. Only Part B1 & B2 are allowed for the Resit this year, because of COVID19; and the projects should be done individually in Resit. === (More information will be given in the first class) disclaimer: Information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	Different conditions for different test/exams. Conditions will be informed 1 week before the exams/test.

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

IN4049TU	Introduction to High Performance Computing	6
Responsible Instructor	Prof.dr.ir. H.X. Lin	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear algebra (matrix and vector operations), numerical analysis (solution of a system of linear equations;some experience with a programming language (e.g., C) is preferred but not required.	
Course Contents	This course is intended for students who are interested in computing-intensive research. In the course, a number of algorithms that are being used within a diversity of research areas is considered. The scaling behaviour of these algorithms in case of an increasing problem size and/or an increasing number of processors, is analysed. Attention is paid to those aspects of computer architectures that are important to understand the resulting performance, such as the memory hierarchy and the interconnection network. By analysing a number of case studies (applications) with respect to their computing-intensive character, possible bottlenecks will be determined. Based on performance analysis, it will be indicated how the effect of those bottlenecks can be reduced. The goal is to learn how to get a high performance with the available hard/architecture. The lab exercises will be done on a cluster of computers, the DAS-5 system at TU Delft. The emphasis will be on designing efficient parallel algorithms and on the necessary optimalisation of the performance. During the lab exercises, the following types of problems will be elaborated on: a parallel Poisson solver, a parallel finite element simulation and a parallel N-body simulation. More information, such as handouts and slides, can be found the Brightspace.	
Course Contents Continuation	High Performance Computing, parallel programming, parallel algorithm	
Study Goals	1. Knowledge about high performance computer systems including parallel and distributed architectures, and programming models; 2. Basic knowledge about the concepts of data decomposition and parallel algorithms; 3. Knowledge about various high performance (numerical) methods and their parallelization; 4. Capable to implement parallel programs (using MPI) on cluster of computers and GPU (using Cuda); 5. Obtain some experience on performance analysis of parallel programs.	
Education Method	Lectures, computer lab exercise using MPI. As an option, answers to the bi-weekly quizzes can be handed in, and a maximum of one bonus point to the exam grade can be obtained.	
Computer Use	Lab exercises (mandatory): implementing (small) parallel programs with C, MPI and Cuda.	
Literature and Study Materials	Will be made available throughout the course and can be downloaded from the Brightspace.	
Assessment	Written exam (50%) + Lab work (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Via Osiris	

IN4191	Security and Cryptography	5

Responsible Instructor
Z. Erkin

Instructor
Dr. K. Liang

Contact Hours / Week x/x/x/x
4/0/0/0

Education Period
1

Start Education
1

Exam Period
1
2

Course Language
English

Required for
UT-201500042 Privacy Enhancing Technologies (Q4)

Expected prior knowledge
Basic understanding on the following is suggested.
-Probability and statistics
-Programming skills

Course Contents
Motivation:
Computers are now found in every layer of society, and information is being communicated and processed automatically on a large scale. Examples include medical and financial files, automatic banking, video-phones, pay-tv, teleshopping and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy.

Synopsis:
Security and cryptography are essential components of any digital system. In this course, the fundamentals of secure data storage and transportation of information are described. In particular, classical (e.g. Caesar, Vigenere) and modern encryption schemes (RSA, DES, AES, Elliptic curves) are described along with their mathematical background such as number theory. Methods for authentication, data integrity and digital signatures are discussed in detail, as these are the main components of many security architectures. The course also investigates more advanced topics such as zero-knowledge proofs and secret sharing schemes.

Aim:
It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security and privacy, as well as is familiar with present applications.

Learning outcomes:
The goal is to make students familiar with the basic concepts applied cryptography, including classical cryptography and modern secret key and public key cryptography. In particular, the students will acquire
A sound understanding of the notion of security
An understanding of the confidentiality, integrity and authenticity needs of the society
Understand the role of cryptographic primitives including the differences between symmetric and asymmetric cryptography, the role of hash functions, digital signatures and PKI
Understand the advanced topics in cryptography needed for the modern society with untrustworthy entities

Among others things, the following topics are covered:
-Classical systems
-Information theoretic security
-Definition of Security notions
-Symmetric encryption (e.g. DES, AES)
-Asymmetric encryption (RSA, Elliptic Curves)
-Hash functions
-Random number generation
-Key Management
-Digital Signatures,
-*Secret Sharing. (if time permits)
-*Zero Knowledge proofs (if time permits)

Study Goals
It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security, as well as is familiar with present applications.

Through assignments, students are expected to have the chance to work on the topics covered in the lectures.

Education Method
Lectures, assignments and weekly exercises.

Attention: This course requires full effort of 140 hours. Even more, if you lack the background (probability and modular arithmetic)

Planned Workload:
Lectures: 28 x 45minutes sessions, total 22 hours
Practice session: 7 x 90 min. total 12 hours
Assignments: 3 x 20 hour, total 60 hours
Weekly study: 7 x 4 hours, total 28 hours
Exam preparation: 20 hours
Exam: 3 hours

Literature and Study Materials
Cryptography made simple, Nigel P. Smart, 2nd Edition, Springer, 2016 (PDF Available Online)

Handouts of lectures

Assessment
Written exam(70%) + mandatory assignments (30%).
Passing grade for the written exam and assignments is required. Please refer to CS regulations for further details.

There is NO reparation for assignments.
Points from previous years cannot be transferred.

NOTE: the study guide information may change depending on the developments around the COVID-19 pandemic

Exam Hours
In case of in person examination at campus:
The exam is closed book.

A cheat sheet of size A4, hand written is allowed for the written exam. Name and student number has to be present on each side.

In case of remote exam from home:
open book: textbook, slides and self-made notes only.
randomised/customised exam.

Permitted Materials during Tests	Only non-scientific calculators.

IN4252	**Web Science & Engineering**	**5**
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4307	Medical Visualization	5
Responsible Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0 lectures & 0/4/0/0 lab.	
Education Period	1 2	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic knowledge of linear algebra, calculus and programming is needed. This course (IN4307) has been designed to complement the courses Advanced Image Processing (ET4283) and Medical Imaging (AP2231TUD). However, these two courses are NOT pre-requisites.	
Course Contents	Theory and practice (Notice project extends to Q2) of medical visualization. This includes the following aspects: data acquisition basics, clinical practice; image processing, e.g., filtering, segmentation and measuremen; medical volume visualization; illustrative visualization; advanced visualization for complex modalities; interaction techniques for medical data; advanced applications.	
Study Goals	By the end of the course, you should be able to LO1: Explain medical visualization algorithms and their applicability to medical problems. LO2: Discuss the advantages and disadvantage of medical visualization algorithms. LO3: Build a medical visualization system for a given problem: a. Discuss a suitable visualization for a given medical problem. b. Implement the most suitable solution. c. Judge the performance of the implemented solution.	
Education Method	The course will be based on a combination of lectures and practical assignments. A final project will be developed in Q2	
Literature and Study Materials	Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications Bernhard Preim and Charl P. Botha (not mandatory)	
Assessment	The evaluation will be based on - a written (or oral if the number of students allows) exam (40%) - assignments during the semester (10%) - a final project (50%) The final project will be done during the 2nd quarter. Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The assignments will consist of small programming exercises and open questions, as preparation for the final project. The practical sessions will provide time for working on the assignments with assistance. The deliverables will be program code and/or answers to questions. The final project will be the design and implementation of a visualization system for a given medical problem. The final project will be carried out in teams. The deliverables for the final project will be a report (paper), the results (e.g., code) and a short video presenting the project (i.e. screencast). The written exam will be arranged at the end of the first quarter. You are allowed to have the slides and material of the course during the exam. No computer or laptop is allowed. The exam has a resit. The project will have a resit if the exam (NOT the resit exam) has been passed with a mark of 7.5 or higher and it will be on an individual basis. The project resit is not automatic and must be initiated by you within two weeks of the grades being published. Resit of a project will mean starting a new project. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	Notes and written material. No computers.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

IN4387	System Validation	5
Responsible Instructor	C.B. Poulsen	
Contact Hours / Week x/x/x/x	4,0,0,0 Lectures & 2,0,0,0 lab	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	Embedded Systems Masters	
Expected prior knowledge	There are no strict entry conditions for this course. However, prior knowledge of requirements analysis is recommended. Furthermore, a good basic knowledge about logic and set theory is extremely beneficial.	
Parts	Behavioural specification of sequential and parallel using labelled transition systems, process algebra, and abstract data types; model checking of such systems using the modal mu-calculus. Model-based testing.	
Summary	Everyone who ever designed an embedded system or a communication protocol involving several components executing simultaneously has experienced that such software is inherently susceptible to bugs. Typical problems that occur are race conditions, deadlocks, and unexpected interplay between different components. Due to the parallel nature of these systems, it is notoriously hard to detect such bugs using testing (for example, timing plays a crucial role). The following quote from the famous Dutch computer scientist Edsger W. Dijkstra illustrates a further problem with testing. Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence. Edsger W. Dijkstra In this course, we study model checking, which in contrast to testing can also be used to show the absence of bugs. Model checking is a technique in which we consider all states in (a model of) the system based on an abstract model. Based on this state space we verify whether the model satisfies the desired properties. Properties are typically derived from the requirements of the system. We will restrict ourselves to verification techniques that do not reason about timing (merely about the order in which event happen). Finally, we see how model-based testing can be used to show that an implementation conforms to the specification of the system.	
Course Contents	Behavioral Specification using Process Theory (Labelled Transition Systems, various notions of behavioral equivalence) and process algebra. Model checking the modal mu-calculus, and model-based testing using IOCO.	
Study Goals	Upon completion of the course: 1. The student knows the fundamental theory necessary for specifying the behavior of embedded systems and for reasoning about this behavior. 2. The student can describe simple systems using this theory. 3. The student can formally specify requirements and prove (or disprove) them on the behavior. 4. The student is able to model a concrete embedded system, and verify that it satisfies its requirements. 5. The student is able to show that an implementation of a system conforms to its specification.	
Education Method	Lectures + Programming Assignments + Practical Project The course is structured into two parts: 1. There will be weekly mandatory programming assignments in the first four weeks of course will be a small set of mandatory. The programming assignments are assessed as pass/fail. The programming assignments are due after the first four weeks of the course. 2. In the last four weeks of the course, you will self-organize into groups of (about) 4 students, and will develop and verify a model of an embedded system. You will write a report that documents your model and its development. There will be a written exam with programming assignments at the end of the course.	
Computer Use	The theory introduced in this lecture is at the heart of the mCRL2 tool set. This tool set can be used to specify and verify systems, and visualize them. To be able to carry out the project it is required that the mCRL2 tool set is installed on your laptop (or one of the TU Delft systems, if you do not have a laptop you can use). It is open source software, and is free of charge. The software can be obtained from https://www.mcrl2.org.	
Literature and Study Materials	The course is based on the book by Groote and Mousavi (see "Books"). All other materials will be published on Brightspace.	
Books	J.F. Groote and M.R. Mousavi. Modeling and Analysis of Communicating Systems. MIT Press, 2014. ISBN: 9780262027717 (Chapters 1-7,11 are mandatory)	
Assessment	The result of this course will be based upon the results of the written examination (50%) and the practical project (50%). For both the programming exam and the practical project, a minimum of 5.0 is required in order to pass the course. To be eligible for taking the exam you must submit and pass the mandatory programming assignments for the first four weeks of the course. Grades of the project or written exam do not automatically carry over from previous years, so upon retaking the course talk to your lecturer first. For the exam a resit is scheduled. Please note that the study guide information for this course may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	The exam will be a 3 hour written exam with programming questions. You are allowed to use the book and any other static resources. You are not allowed to communicate or discuss exam questions with anyone but members of the teaching team for the course. Discussing or copying code will be considered fraud, and is reason for expulsion from the course.	
Enrolment / Application	Brightspace	
Co-Instructor	E. Visser	

WM-ITAV-4010	Scientific Writing	2
Module Manager	L. Meester	
Instructor	A. Glasbergen-Plas	
Instructor	M. Looij	
Instructor	M.J.Y. Wackers	
Instructor	S. Baars	
Instructor	M. Bliekendaal	
Instructor	L.C. Schroten	
Instructor	Drs. W.J. Blokzijl	
Instructor	Drs. B.M.D. van der Laaken	
Instructor	Drs. P.C. Post	
Instructor	Drs. A.E. Kam	
Co-responsible for assignments	Drs. B.M.D. van der Laaken	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1 2	
Exam Period	none	
Course Language	English	
Course Contents	In this course, you learn to write a scientific article, either a research article based on your own research data or a literature review about a subject of your own choice. This is a necessary skill for anyone who wants to pursue an academic career after their graduation, but it can also be used immediately for all academic texts you will write during your Master programme, such as your Master thesis. In seven weeks, we will go through all steps of the writing process, from formulating a good main question and finding relevant literature, to the actual writing, re-writing and final editing. You will exercise with finding, reading and managing relevant academic literature, writing in an academic style, building a comprehensive argumentation, reviewing fellow students' articles and using other students and the instructor's comments to improve your own work.	
Study Goals	The purpose of this course is to learn how to write a scientific text. To achieve this, at the end of this course you will: know what the main characteristics are of a scientific text be able to formulate a main question be able to find, critically read and manage scientific literature be able to use literature properly and avoid plagiarism be able to build up your argumentation be able to structure an article according to the conventions in your field of study be able to use scientific English style be able to use tables and figures to support and communicate your results be able to give feedback on somebody elses article be able to use feedback for improving your work.	
Education Method	(Online and/or on campus) practical, in 6 sessions (attendance mandatory). Every week you have to read some background information about scientific writing and hand in a part of your text. Participants must attend all sessions - one missed session is allowed only - and hand in all assignments in time. Students who receive a pass for this course are rewarded with 2 ects. This equals 56 hours of study. A total of 12 hours is spent on attending (online) classes, in which you can ask questions, discuss the feedback you received on your work and discuss aspects of scientific writing with your fellow students; the remaining 44 hours is for self study, writing and revising. In seven weeks, from preparing lecture 2 up to handing in your final article in week 8, you will have to spend at least 6 hours of self study on this course, every week. It is important that you make sure you have this time in your personal schedule. At the beginning of the course you will mainly be reading up about scientific writing and the subject of your text. As the course proceeds, you will be spending more of your time on writing, giving feedback and revising your own text.	
Books	Theory about academic writing will be made available through Brightspace.	
Assessment	You write a scientific article of 3000 words (excluding the list of references and the abstract, 10% margin in the word count). You have to hand in a parts of your article every week. Your final grade is based on the final article. An evaluation form for the grading of the article is available throughout the course.	
Elective	Yes	
Category	MSc level	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 2 2021

CS4015	Behaviour Change Support Systems	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	none	
Course Contents	Behavior change support systems (BCSS) are computer-based systems that support individuals to form, alter or reinforce cognitions, attitudes or behaviors without using coercion or deception. They can serve individuals throughout the various stages of a change process, such as awareness developing, contemplation, action strategy development, development of new behaviors, and maintaining these new behaviors. Virtual healthcare coaches, negotiation support systems, and applications that provide individuals with personalize financial guidance are three examples of these systems. To establish, modify or maintain change BCSS can deploy computerized persuasive strategies (e.g. reducing effort to establish target behavior, or argumentation and reflection strategies), simulations (e.g. serious gaming, virtual reality), relational software agents (e.g. ePartners, virtual coaches), and personalization based on longitudinal user data. BCSS are found in many domains, including education, sales, negotiation, management, and particular in the health domain.	
Study Goals	The course allows students to achieve understanding of principles, concepts and theories underlying BCCS systems and methods for designing them.	
Education Method	In the pre-recorded video material, theories, principles and methods are presented, discussed and illustrated with examples from the field. The video material is support by online self-tests. In the lectures, teacher and students discuss and make assignment related to pre-recorded material of that week. At home students work on their own in small groups on coursework assignments to develop a product design for a BCSS. In the practicum session student groups presented the progress on their coursework and receive feedback. Expected Workload Pre-recorded video material: 18 hours (2 hours × 9 lectures) lecture: 10 hours (10 × 1 hours) practicum 14 hours (7 × 2 hours) Reading time: 36 hours (9 × 4 hours reading time) Preparation presentation: 3 hours (3 × 1 hours for each presentation) Coursework project, including writing report, and final presentation: 50 hours Exam preparation and revision: 9 hours Total = 140 hours	
Literature and Study Materials	Will be announced on brightspace	
Books	Wendel, S. (2013). Designing for Behavior Change: Applying Psychology and Behavioral Economics. " O'Reilly Media, Inc.".	
Assessment	The course is assessed by coursework and an exam as follows: (60%) computerised examination (or oral exam) (40%) Coursework Project (resulting in a report, and final presentation include question and answer round where individual group members are assessed on coursework) If the expected number of students registering for exam or resit is small, the teacher might decide to replace the computerized examination by an oral examination. Separate marks will be given for exam and coursework, only a combined mark is recorded in Osiris. A passing final grade for the course can only be earned when for both the exam and coursework at least a 5.0 is earned, and the weighted grade for exam and coursework is at least a 5.8. Resit coursework A second submission of modified coursework is only allowed for coursework that received a fail mark (<5.8) for the first submission. Overall resit mark for coursework will be capped to 5.8. Note that individual marks for coursework or computerised exam (or oral exam) do not carry to the next year. disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	2	
Co-Instructor	M.L. Tielman	

CS4055	High Performance Data Networking	5
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Basic understanding of networking and programming (ideally Python).	
Course Contents	The Internet has become of critical importance to society. However, the large size of networks and abundance of protocols have made network management very complex. The novel concept of network programmability addresses this complexity and has resulted in a paradigm shift in how networks are (or can be) operated. The high-performance data networking course is an advanced networking course that will introduce you to the concept of network programmability and which treats fundamental networking concepts like Quality of Service and network resilience.	
Study Goals	The learning objectives of this course are twofold: (1) The student should gain knowledge of the treated networking technologies. (2) The student should be able to apply and work with the programmable network technologies in a network emulator (Mininet).	
Education Method	Approximately 50% of the course will consist of lectures and selfstudy and 50% focuses on (homework) exercises and instruction classes.	
Literature and Study Materials	Slides and a reader containing the exercise material.	
Assessment	The final assessment will be based on an exam that covers both the theory from the slides as well as the content from the reader.	
Remarks	Disclaimer: The information about CS4055 (including its assessment) may change depending on the developments around the coronavirus.	

CS4090	Quantum Communication and Cryptography	5
Responsible Instructor	Dr. S.D.C. Wehner	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear Algebra, Probability & Statistics, Q101 (Fundamentals of quantum information)	
Course Contents	This class will introduce you to the fascinating field of quantum communication! We will look at the state of the art of quantum networks, and explore techniques for building quantum repeaters that promise to deliver qubits over long distances. We also briefly look at one of the most famous application of quantum cryptography, quantum key distribution. Caution: 1. This class requires you to take "Fundamentals of Quantum Information" in Quarter 1 2. The focus of this class is presently on quantum communication, and we will only briefly look at quantum cryptography. As such, this class is not held in flipped classroom style in conjunction with edX QuCryptoX as in previous years.	
Study Goals	The student will acquire: A good understanding of the fundamental concepts of quantum communication Insight into the differences between classical and quantum communication and cryptography Skill set required to follow the remainder of the quantum curriculum (Q301 Quantum hardware and Q401 Quantum electronics)	
Education Method	Lectures and tutorials. If remote classes continue this fall, then recorded lectures and live discussion session	
Literature and Study Materials	Primary: Slides Review Articles Auxilliary: Nielsen and Chuang Quantum computation and information, Cambridge University Press.	
Assessment	Homework (70%) and Final Project (30%) disclaimer: information may change depending on the developments around the coronavirus.	
Tags	Abstract Adventurous Algorithmics Challenging Group Dynamics/Project Organisation Information & Communication Integrated Intensive Involved Lineair Algebra Mathematics Physics Quantum Signals Technology Telecommunication	

CS4200-B	Compiler Construction B	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/6/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	- CS4200-A: Compiler Construction A (recommended) - programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Note that the title of this course should be "Compiler Construction B", not "Compiler Construction Project". The course combines theory and practice of compiler back-ends. Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-B covers the following topics: * Transformation - rewrite rules, rewrite strategies - simplification, desugaring * Dynamic semantics and code generation - operational semantics, program execution - virtual machines, assembly code, byte code - code generation - memory management, garbage collection * Data-flow analysis - control-flow, data-flow - monotone frameworks, worklist algorithm	
Study Goals	After this course students students should be able to * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define transformations on abstract syntax terms to simplify programs * Define a code generator that translates source language abstract syntax trees to object language instructions using rewrite rules * Define data-flow analyses using control-flow and data-flow rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the back-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%)	

	- course project (50%) Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	not applicable
Judgement	to be decided

CS4220	**Machine Learning 1**	**5**
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

IN4089	Data Visualization	5
Responsible Instructor	T. Höllt	
Instructor	Prof.dr. E. Eisemann	
Contact Hours / Week x/x/x/x	0/2/0/0 & lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	Master course MKE/ST/DS	
Expected prior knowledge	IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required). The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.	
Course Contents	Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization, The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines. As a computer science course, affinity to algorithmic thinking and programing skills will be needed. Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.	
Study Goals	The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data. Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more. By the end of the course, you should be able to LO1: Discuss a large range of visualization techniques. LO2: Discuss a perception principle of visualization. LO3: Explain mathematical principles and algorithms of visualization techniques. LO4: Design suitable visualization systems for a given practical data analysis problem. LO5: Implement visualization systems for a given practical data analysis problem.	
Education Method	Lectures, practical assignments, self-study, and projects.	
Literature and Study Materials	Course slides, instructions for projects, and selected literature. Chapters from: Visualization Analysis and Design Author: Tamara Munzner CRC Press Visual Computing for Medicine 2nd Edition Theory, Algorithms, and Applications Authors: Bernhard Preim Charl Botha Morgan Kaufmann All available in electronic form via Brightspace or at TUDelft library.	
Assessment	The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Judgement	The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam. The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project. Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions. The project is evaluated based on the developed result, its documentation and presentation. Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.	

IN4150	Distributed Algorithms	6
Responsible Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Knowledge of Computer Networks (CSE1405) and Operating Systems (CSE2430) is useful as background for understanding this course.	
Course Contents	Introduction to distributed algorithms; notions of time and ordering of events; distributed algorithms for message ordering, detecting global states, termination detection, deadlock detection, mutual exclusion, election, minimum-weight spanning trees, fault tolerance, consensus, and agreement; blockchain technology and its relation with consensus.	
Study Goals	Understand the main fundamental problems in distributed systems Understand the most important distributed algorithms that solve these problems Be able to reason about the execution of distributed algorithms Be able to program distributed algorithms Be able to select and summarize relevant literature on distributed algorithms	
Education Method	Lectures and lab work executed in groups of two students	
Literature and Study Materials	Lecture notes and lecture slides (available on Brightspace)	
Assessment	One paper summary and an in-person, on-campus written exam (closed book), which together are an integrated whole. The grade of this exam, which includes the grade of the paper summary (with a weight of 25%), is the grade for the course. In case in-person, on-campus written exams are not possible because of COVID-19, the written exam will be replaced by individual online oral exams.	
Permitted Materials during Tests	None except the list of algorithms	
Remarks	Lab work is 40 hrs.	

IN4302TU	Building Serious Games	5
Responsible Instructor	Dr.ir. A.R. Bidarra	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2	
Course Language	English	
Expected prior knowledge	For CS students: programming experience with some object-oriented language; experience with graphics, AI and/or some game engine(s) is a plus. For all students: though not compulsory, it may be convenient to have followed the course SEN9235 (Game Design Project), which is taught in the first quarter.	
Course Contents	Project-based interdisciplinary course, open to MSc students of all faculties. The main goal of the project is to take students with varying talents, backgrounds, and perspectives and put them together to do what none of them could do alone: to design and implement a serious game aimed at being applied in a real-world setting (educational, social, training, health-related, etc.). The emphasis is both on constructively fulfilling the game requirements, and on deploying the adequate technology for that purpose. Assignments for this course will be provided by real-world end-users (e.g. companies or the Science Centre Delft), to whom the group will be reporting throughout the term of the project.	
Study Goals	At the end of the project, the student will demonstrate proficiency in the following aspects: o identifying and valuing the soft skills necessary to work in interdisciplinary teams o responsibly interacting within a team, integrating its members' varying talents and expertise o adapting with flexibility to the dynamic requirements of a complex external assignment o translating feedback received into proactive personal development steps Additionally, the CS student will demonstrate proficiency in the following specific aspects: o identifying, selecting and deploying the most adequate game technologies for the given serious game domain and constraints o deepening programming skills while building a complex and large software system in an agile context	
Education Method	Project: teams work intensively as a small game studio. Also a few plenary sessions and/or lectures (though less likely in 2021/22).	
Assessment	Project assessment will be based on a combination of: - (~50%) product grade: unique for the whole group, based on both the game itself and the required documentation; - (~45%) process grade (individual), including personal contribution, performance, attitude, and peer evaluation; - (5%) final presentation. The commissioner will be involved both as advisor and as assessor. The final documentation will include writing a scientific paper and actually submitting it to a conference on serious games and/or their application. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr. R. Marroquim	

IN4341	Performance Analysis	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	0/2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	This course applies probability theory and the theory of stochastic processes to the design and performance evaluation of complex networks such as man-made networks as telecommunication, computer and embedded networks and biological networks. The computation with random variables is reviewed. Markov processes and queuing theory will be introduced to the current important concept of "Quality of Service (QoS)" provisioning and to the computation of the blocking probabilities in telephony (both fixed as mobile). Several applications (e.g. the robustness of networks, epidemics in networks, the Internet shortest path routing) are also included. More details are found on brightspace.	
Study Goals	The course intends to provide students with mathematical techniques, in particular probabilistic methods and graph theory, to compare the performance of different network designs and protocols.	
Education Method	Lectures and homework after each class	
Literature and Study Materials	We follow the book Performance Analysis of Complex Networks and Systems, by P. Van Mieghem, Cambridge University Press (2014). See http://www.nas.ewi.tudelft.nl/people/Piet/bookPA.html	
Assessment	Written and closed book. A formularium is provided that can be consulted at the examination. disclaimer: information may change depending on the developments around the coronavirus.	

Specialistievakken start kwartaal 3 2021

AP3132	Advanced Digital Image Processing	6
Responsible Instructor	Prof.dr. B. Rieger	
Instructor	Dr. F.M. Vos	
Contact Hours / Week x/x/x/x	0/0/4/2	
Education Period	3 4	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Basics of signal processing, image processing, linear algebra, elementary statistics.	
Course Contents	The course Advanced Digital Image Processing covers the principles of several state-of-art image processing techniques. Particularly, students will study the theory of sophisticated algorithms for: 1. Multi-resolution Image Processing: gaussian scale space, windowed Fourier transform, Gabor filters, multi-resolution systems (pyramids, subband coding and Haar transform), multi-resolution expansions (scaling functions and wavelet functions), wavelet Transforms (Wave series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform (CWT), Fast Wavelet Transform (FWT)); 2. Morpological Image Processing: advanced operations for binary morphology; definitions of gray-scale morphology regarding erosion, dilation, opening, closing; application of gray-scale morphology including smoothing, gradient, second derivatives (top hat) and morphological sieves (granulometry); 3. Image Feature Representation and Description: measurement principles: accuracy vs. precision ; size measurements: area and length (perimeter); shape descriptors of the object outline: form factor, sphericity, eccentricity, curvature signature, bending energy, Fourier descriptors, convex hull, topology; shape descriptors of the gray-scale object: moments, PCA, intensity and density; structure tensor in 2D and 3D: Harris Stephens corner detector, isophote curvature. 4. Motion and optic flow: Taylor expansion method; dual and multi-frame image registration, optic flow; 5. Image Restoration: Noise filtering, Wiener filtering, inverse filtering, geometric transformation, grey value interpolation; 6. Image Segmentation: thresholding, edge and contour detection, data-driven segmentation (boundary detection, region-based segmentation, watersheds, graph-cut, meean shift), model-driven image segmentation (Hough transform, template matching, deformable templates, active contours, ASM/AAM, level sets).	
Study Goals	General learning objectives of the course are: 1. Student has knowledge of can explain the function of state-of-the-art image processing algorithms; 2. Student can solve elementary problems in image processing using Python/MATLAB? programming; 3. Student can solve more advanced problems without implementation, but sketching steps towards a solution; 4. Student can independently acquire new knowledge about image processing from the current literature and present and report about it.	
Education Method	Lectures, practicals and group assignment with plenary presentation and discussion.	
Computer Use	Matlab including the dipimage toolbox and/or other image processing toolboxes.	
Literature and Study Materials	Book 'Digital Image Processing', van R.C. Gonzalez en R.E. Woods, third edition, 2002, ISBN 9780131687288. (Online) Book 'Computer Vision, Algorithms and Applications', R. Szeliski, (http://szeliski.org/Book/). The online version is available for free. We have used the Book Introductory Techniques for 3-D Computer Vision, E. Trucco and A. Verri, ISBN 0-13-261108-2 in the past. Lecture notes Fundamentals of Image Processing (http://homepage.tudelft.nl/e3q6n/education/et4085/sheets/ppt/FIP2.2.pdf) PDF-files of the lecture slides (see Brightspace).	
Assessment	Closed book written exam and assignment. Both parts should be graded 5.8 or higher. A bonus point of 1.5 (to the exam) can be obtained by attending the practicals with 6 out of 8 passed. The final grade is the average of the two parts. The formula for the final grade is: ((0.85*EX + 0.15)+ AS)/ 2 or without the bonus point from the practicals: (EX+AS)/2 With EX the exam grade and AS the grade for the assignment. If you have not passed the exam or the resit, you will need to redo the assignment again next year!	
Permitted Materials during Tests	Closed book exam; books, print-out of pdf files of the lecture slides and lecture notes are not permitted during the written examination.	
Elective	Yes	
Tags	Image processing Matlab Physics	

CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
Responsible Instructor	Ir. S.E. Verwer	
Responsible Instructor	Dr. A. Panichella	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software is one of the most complex artifacts of mankind has ever created, but complexity is the enemy of correctness. Modern software testing and validation tools use a multitude of techniques geared toward correct computer code, most of these are base on artificial intelligence. In this course, we study these techniques in details, specifically we will understand and implement: Execution monitoring and taint analysis Branch distance computation Hill-climbing and genetic algorithms Concrete and symbolic (concolic) execution Active state machine learning Genetic programming The goal is to better understand and test software using artificial intelligence. Using the taught techniques you will be able to automatically: Discover which code is reachable Find (security) bugs in software Write tests that cover all reachable code Reverse engineer a code's functionality Patch code to remove bugs and failing tests	
Study Goals	The student will: Understand modern AI techniques for software testing. Be able to implement several such techniques from scratch: - smart fuzzing (probing software with input to find crashes/bugs), - symbolic execution (using logic to construct inputs that trigger specific code branches), - fault localization (given that a program fails, find the line of code responsible for the failure), and - automated program repair (using a patch library and genetic programming to improve code) Be able to apply this technology to locate bugs in real-world software implementations.	
Education Method	The main part of the course will consist of 3 lab assignments covering the theory (fuzzing&tainting, symbolic execution, automated program repair), and one lab assignment for the application to real software. The students will implement the taught techniques from scratch in the first 3 assignments, which will be scored with a pass/fail. All three assignments need to be passed to complete the course. The final lab will contain a recap from the first three assignments and an application of a state-of-the-art tool on real software. The final lab will be graded and be the final course grade. There will be instruction sessions where students can work on their assignment and ask the teachers for assistance.	
Assessment	First three lab assignments (pass/fail). Final lab (100%).	
Tags	Artificial intelligence Software	

CS4135	Software Verification	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Ir. K.F. Wullaert	
Contact Hours / Week x/x/x/x	0/0/2/0 + 0/0/4/0 practicum	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	How can we ensure that software cannot crash and is guaranteed to be correct? In this course we tackle this question by viewing programs and programming languages as mathematical objects. That way we can use logic to prove properties about programs and thereby guarantee that software is correct. To make reasoning about actual programs and programming languages feasible, we will not be doing these proofs by hand, but instead use a tool called a proof assistant to build proofs that can be checked by a computer. As we will show during this course, proof assistants turn the activity of doing proofs into programming. This course assumes familiarity with functional programming and elementary logic. This course is a specialization course for programming languages and software engineering	
Study Goals	After this course students will be able to: - State and prove properties of functional programs in logic. - Specify the semantics of a programming language in logic. - State and prove the correctness of imperative programs. - Use a proof assistant to perform a mechanized proof.	
Education Method	This course consists of a weekly lecture of 2 hours and a lab session of 4 hours. During the lab sessions students will work on proving simple theorems. Towards the end of the course students will carry out research projects that apply the ideas of the course.	
Literature and Study Materials	Supplementary material: Free online text book "Logic and Proof": https://leanprover.github.io/logic_and_proof/ Free online text book "The Hitchhikers Guide to Logical Verification": https://github.com/blanchette/logical_verification_2021/raw/main/hitchhikers_guide.pdf	
Assessment	The final grade consists of the following parts: - A programming project in a proof assistant. - A written exam Both have weight 50% and both should be 5 or higher. The weighted average should be 5.8 or higher. The research project should be done individually.	
Co-Instructor	E. Visser	

CS4160	Blockchain Engineering	5
Responsible Instructor	Dr.ir. J.A. Pouwelse	
Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	In this course you will learn all aspects of blockchain technology, including tamper-proof data structures, digital identities, transitive trust, fault tolerance, distributed consensus, smart contracts and applications. Ledgers and blockchains are an emerging technology with the potential to radically improve financial transactions, supply-chain flows, transactions in general, and distributed databases. The first three weeks of the course will provide a fast-paced introduction to Bitcoin, Ethereum, and TrustChain developed by TUDelft itself. The main component in this course is a team-based complex engineering project. This course is designed for computer scientists to understand blockchain technology and to produce significant hands-on experience. To provide a deep understanding of blockchain technology and understand why it is special you need to experience first-hand how it operates at a detailed technical level. Students design, implement, and test their own independent project in teams of 3-5 students. Students can choose from a pool of possible project ideas. This course requires you to like software engineering. Topics covered: -Blockchain basics and evolution Bitcoin 1st generation, smart contract generation, future 3rd generation (trust or trust in math) -identity and transitive trust Authentication and security primitives, tamper-proof identities, trust models, MITM attacks, Sybil attacks, and TrustChain by TUDelft -Consensus models Proof-of-work, permissioned, Proof-of-stake, Corda no-global-consensus, TUDelft bottom-up fast consensus model -Smart Contract pro/con encrypted data, Bitcoin scripts, Etherium execution model, Hyperledger + Docker issues, Corda Jar file approach, Tezos difficult to use, powerful technology, vision of the future: trusted verified execution -Markets and exchanges Disruption by open markets, winner-takes-all, and multi-sided market platforms, Uber, Airbnb, 22 years of eBay, Silk Road, honesty among drug dealers, the role of trust in markets, P2P exchange markets -Decentralized Autonomous Organization, novel method to collaborate and organise any economic activity Within this 2021 edition "the Delft DAO" will be prominently featured. TUDelft achieved a world-first in DAO research. We devised a full end-to-end proof-of-principle of a DAO which is capable of 0) near unbounded scalability 1) controlling money 2) democratic decision making and 3) continuous sustained self-evolution. This course provides you with the knowledge to work with this advanced technology. After this course you will have a firm grasp on the current operational blockchain-based systems, realistic view of real-world applications that may be built on top of ledger technology. You will be able to reason and discuss the open challenges and questions that still need to be resolved. This course is a key course for distributed systems students.	
Study Goals	After this course students are able to design and engineer complex blockchain-based systems. Students are able to describe blockchain technology, the various consensus model, smart contracts, markets, and relation to existing database technology. Student are able to setup a new architecture for blockchain applications.	
Education Method	This course consists of four 2-hour lectures. Each lecture is followed by a 4-hour homework period in the same week focused on understanding the background material. In week 1 you will form teams and initiate work on your blockchain engineering project. A list of projects to select from will be provided at the start of this course.	
Literature and Study Materials	Online course textbook: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Narayanan, Bonneau, Felten, Miller and Goldfeder.	
Prerequisites	It is highly recommended to follow this course (see remarks): Security and Cryptography (Q1) Distributed Algorithms (Q2)	
Assessment	The final grade reflects the quality of your work and team cooperation. This course has a minimal amount of formalities. You will write down your project results in a single-page report, IEEE style. You will be graded on your open source efforts located on Github and single-page report. Your grade will be expressed on a scale of 0 to 10. (resits or repair options are not offered for this course) Covid-19 disclaimer: the assessment and course format could be altered at any time !!!	
Remarks	This class has a limited capacity (50). If there is a larger number of enrollments than the capacity of the class, students will be assigned to their preferred blockchain engineering project based on their background, engineering experience level, and match to the course goals. Students who followed Security and Cryptography (Q1) and are also enrolled in Distributed Algorithms (Q2) will have priority for placement. Mathematics students are exempts from this, if they can show some minimal software development experience (e.g. Github profile). Finally, students with a Grade Point Average of 8.0 or higher are eligible for the challenging scientific projects, resulting in a research paper. These project receive intense guidance, but have no capacity limits.	

CS4195	Modeling and Data Analysis in Complex Networks	5
Responsible Instructor	H. Wang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The assignment and final project require basic programming skill.	
Course Contents	Big Data is mostly obtained from features of components and the interactions among components in large complex systems. Examples are (1) end user features and interactions in both online and real-world social networks like Twitter, LinkedIn (2) data from content sharing platforms such as YouTube (3) physiological data of the brain and (4) stock prices etc. in economic systems. Such a dataset is networked in nature i.e. the data of the system components or interactions are (cor)related to each other. This course introduces the basic methodologies to analyze, model, interpret and possibly to predict such Networked Data, combining advances from network science, modeling of dynamic processes and statistical physics, beyond machine learning algorithms. These methods will be applied to diverse real-world datasets obtained from e.g. Facebook, LinkedIn, YouTube, the brain etc.	
Study Goals	After this course, students could construct a network based on the dataset, characterize and model the network in order to e.g. detect patterns and anomalies, model the data via dynamic processes (e.g. viral spreading) on networks to decode the underlying governing mechanisms of e.g. information/error/behavior contagion and to predict e.g. the popularity of a product, news, disease, computer virus, control the contagion process such as maximize the information prevalence and market share. Students could obtain an overview of the Msc/Phd projects on the frontiers of networked data analysis.	
Education Method	In total, there will be about 7 lectures. Students will also learn via an assignment and a final project (each group gets individual supervision).	
Assessment	The final grade is based on the assignment (20%) and final project (80%). There is no resit for both the assignment and the project. disclaimer: information may change depending on the developments around the coronavirus.	

CS4210-A	Algorithms for Intelligent Decision Making	5
Responsible Instructor	Dr. N. Yorke-Smith	
Instructor	Dr. E. Demirovi	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Recommended: IN4010: Artificial Intelligence Techniques, or equivalent; and/or IN4301: Advanced Algorithms, or equivalent Required: basic course(s) in algorithm design and analysis, logic and probability; basic programming (in Python)	
Course Contents	Decision making is at the centre of artificial intelligence. This course gives you practical skills on a solid theoretical base. The course looks at solving mathematical models of NP-hard discrete optimisation problems. These kinds of problems lie at the heart of AI techniques such as planning, machine learning and mechanism design, and more generally combinatorial optimisation. You will learn about a range of modelling techniques from boolean satisfiability to constraint programming, and how advanced solvers for these models work. The course has plenty of real-world case studies as well as theoretical results. Apply the skills you learn in this course by taking CS4210-B: Intelligent Decision Making Project in quarter 4!	
Study Goals	By the end of this course, you will be able to identify features of real-world combinatorial decision problems, and be able to model and design systems for simplified instances of these problems using boolean satisfiability, mixed integer programming, and constraint programming over finite and real domains. You will be able to explain how SAT, CP and LCG solvers work in some detail, and how MIP solvers work at a high level.	
Education Method	Lectures, homework exercises (optional), and programming assignments. The expected workload is: 30% lectures (including preparation for the exams) 40% homework exercises (optional) 30% programming assignments	
Literature and Study Materials	Provided on Brightspace	
Assessment	The final grade depends on the grades obtained for (a) programming assignments (2 in total) [30%] and (b) the exam [70%]. The final grade is computed from the unrounded grades for the components. The final grade for the programming assignment is a uniformally-weighted average of the unrounded grades obtained for the two assignments. Programming assignments can be completed by two students working together. The exam is graded on a scale from 1 to 10. A resit will be available for the exam. The result for the exam is determined by the maximum score obtained for the original exam and the resit. In order to pass the course, the rounded grade (after resit if applicable) for each part of the course must be at least 5.0, and the rounded final grade on the course must be at least 5.8. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Algorithmics Artificial intelligence Group work Modelling Optimalisation Programming Projects Small groups	

CS4225	Educational Technologies	5
Responsible Instructor	Prof. M.M. Specht	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	* Theories of Human Information Processing and Learning * Learning Management Systems * Learning Analytics * Personalisation and Adaptive Educational Systems * Mobile and Seamless Learning Technologies * Artificial Intelligence in Education * Realtime Learning Technologies * Project Design * Project Implementation	
Study Goals	The course will enable you to classify, understand, design and implement the core functionalities and systems for supporting human learning processes. As well current practices implemented as also approaches for technology enhanced learning currently researched will be presented. You will learn how educational technologies provide human learning process support, implement guidance and recommendation, create personalised learning support, as also give real-time feedback and support reflection of learners. In the final project you will identify a problem, design a solution based on the presented approaches and implement your own educational technology solution.	
Education Method	Lectures, weekly assignments and quiz questions, final project	
Assessment	Weekly assignments 30%, Final project 70% disclaimer: information may change depending on the developments around the coronavirus.	

CS4230	Machine Learning 2	5
Responsible Instructor	M. Loog	
Instructor	Dr.ir. J.H. Krijthe	
Instructor	Dr. F.A. Oliehoek	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/4/4	
Education Period	3 4	
Start Education	3	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course is the more advanced and research oriented follow-up to CS4220 [Machine Learning 1]. The content of the latter is, therefore, expected as prior knowledge.	
Course Contents	The course will treat a number of machine learning theories and techniques in detail and on an advanced level. Possible topics : - learning theory - Bayesian networks - online learning - Rademacher complexity - Markov decision processes - semi-supervised learning - reinforcement learning - active learning - causal reasoning and discovery	
Study Goals	After successfully completing the course, the student is able to apply the techniques and theories that have been covered in the course. In addition, they are able to develop learning strategies for new and previously unseen situations. Moreover, the student can provide reasoned justifications for these strategies based, for instance, on theory and/or experiment.	
Education Method	Lectures + Q&A sessions	
Assessment	Grading is based on two parts. Following the lectures -- we have about 11 of those, there is an individual assignment that will be graded pass/fail. In addition, there is a written examination that will be graded on a scale from 1 to 10. You pass the course when you both have a pass for the assignment and a passing grade for the written exam. Upon passing the course, your final grade will be the grade for the exam. Finally, note that there is a resit for the written examination, but not for the report. disclaimer: information may change depending on the developments around the coronavirus.	

CS4235	Socio-Cognitive Engineering	5
Responsible Instructor	Prof.dr. M.A. Neerincx	
Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Basic prior knowledge on human-computer interaction is helpful, but not required.	
Course Contents	Whether you are playing a game in virtual reality, driving a semi-autonomous car, educating yourself in a MOOC, or harmonizing your health and lifestyle via a mobile app; nowadays intelligent networked information and communication technology is omnipresent. This course focuses on the design of human-aware intelligence into such environments, to support joint human-technology performances that bring about positive human experiences (such as social robots that help activity coordinators guide health-promoting games for people with dementia, http://rejam.tudelft.nl). In the Socio-Cognitive Engineering (SCE) course (MSc level), you will become acquainted with the application of a coherent set of methods for the design and evaluation of human--agent collaboration. Based on the SCE-method, we will elaborate on the state of the art of intelligent user interfaces (ePartners), such as artificial personal assistants, artificial team mates, eCoaches, social intelligence, and companion agents. The main topics of study are: - Design methods: Cognitive Engineering, Value Sensitive Design, Scenario-based Design, Claims Analysis, Design Rationale, Design Patterns. - Design for collective intelligence: Knowledge Representation, Ontology Engineering, Mental Models, Theory of Mind, ePartners, Adaptive Automation, Socially Intelligent Interfaces. - Design Evaluation: Prototyping, Test Methods, Measures, Questionnaires, Ethics. - Human Factors Theories and Models: Human Cognition & Learning, Memory, Emotion, Task Load, Human-Agent Teamwork, Behavior Change and Persuasive Technology.	
Study Goals	At the end of this course, students will be able to: 1. Explain the essential concepts of the design methods addressed in the course. 2. Explain the (dis)advantages of various design methods and their complementarity. 3. Apply the design methods addressed in the course in their research and design projects. 4. Explain what a design rationale is. 5. Construct a design rationale. 6. Create design specifications that are grounded in a design rationale. 7. Evaluate the strengths and weaknesses of a design rationale, e.g. using human-centered evaluations that test the design rationale. 8. Explain some of the state of the art human factors theories, models, and methods relevant to intelligent user interfaces, human-agent collaboration, and ePartner technology. 9. Write a structured report about a design-test cycle, with sufficient detail for a new group of researchers to continue the research. 10. Present work on a design project to an academic audience. 11. Work in a group on collaborative assignments.	
Education Method	LECTURES During the lectures, the teachers will present a range of theories, models, and methods relevant to socio-cognitive engineering. Students are required to read a number of scientific papers which are made available on Brightspace, along with the sheets/slides of the lectures. Together, the sheets/slides and the papers provide the students with the required theoretical knowledge to work on the practical project, and to learn about relevant design methods, human factors theories, conceptual solutions, and design principles. Most of the lectures include practical assignments and discussions stimulating the students to apply the contents of the lecture to their project (also see Project). PROJECT In the project, students work in groups to apply the knowledge acquired during the lectures. Students are required to plan, execute, present, and report on a complete design cycle (i.e. design, prototype, and evaluation) for a given design problem. This year (like the past years), the design problem is a social robot for older adults with dementia, and their social environment (https://rejam.tudelft.nl). The objective of the social robot is to improve humans physical, social, cognitive, and emotional well-being. The students will use the Wiki Socio-Cognitive Engineering (WiSCE) tool to specify the design rationale and its evaluation, step-by-step (see also https://scetool.ewi.tudelft.nl). Throughout the course, students will give presentations about their progress, on the design and evaluation of their prototype.	
Literature and Study Materials	Papers from scientific journals on Brightspace. Lecture notes on Brightspace.	
Assessment	Literature and study material consist of: - Papers from scientific journals on Brightspace. - Lecture notes on Brightspace The module assessment concerns the processing and application of the theory and methods; the construction of the design (rationale) and the evaluation; and the provision of the resulting concise and coherent report (including the lessons learned): Presentations (10%) Prototype (10%) Project report according to the prescribed format (70%) Individual reflection (10%)	
Exam Hours	There is no exam. The assessment is based on a paper, presentation and report. During the course, students will receive feedback on interim work. There is no resit after the end of the course.	

CS4240	Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	Seminar Computer Vision by Deep Learning	
Expected prior knowledge	Basic pattern Recognition, Basic Machine learning, Basic statistics, Basic probability theory, Programming experience (python + numpy). Note: Without some affinity/experience with python (and numpy) programming it might still be possible to pass the course, but will become quite difficult.	
Course Contents	In this course we will look at a specific field of Artificial Intelligence and Machine Learning: Deep learning. Deep learning has shown remarkable success with large data sets and unstructured input data such as raw images/audio/text. Topics include: feed forward networks, back-propagation, optimization, convolutional nets, recurrent nets, self-attention, unsupervised methods. The course will have lectures, a seminar, a lab practical and a project: - The lectures will be on generic topics, following the book; building the backbone. - The lab assignments will have you apply basic concepts of the lecture in python notebooks. - The seminar will have students read, critique, and present relevant deep learning research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The project will have students apply and design their own (small) deep learning project in the context of scientific reproduction.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the different deep learning techniques reviewed in the course, such as SGD, MLPs, CNNs, RNNs, GANs. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [LO4]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (eg: Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) problem domain which may need to re-implement, run, evaluate, investigate, extend existing research or code [LO8] Write clearly and concisely about your code, method, results, and analysis.	
Education Method	Lectures for basic theory based on the literature Assignments: we help you to become familiar with PyTorch; applying concepts from the lecture on small problems. Lab project: design and execute your own deep learning project in the context of scientific reproductions using https://reproducedpapers.org/ Seminar: paper reading, critiquing, and presenting.	
Literature and Study Materials	Books: freely available online: - http://www.deeplearningbook.org/ - https://d2l.ai/ Research papers will be made available through Brightspace. Assignments are based on PyTorch: https://pytorch.org/	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Project: in a small group of students you work on a deep learning paper reproducibility project. 3. Exam about the papers, assignments, and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4275	Web Programming Languages	5
Responsible Instructor	E. Visser	
Instructor	Dr.ir. D.M. Groenewegen	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	Languages and frameworks for web programming are constantly evolving. Over the past decade, there has been a large shift from applications with server-side rendering of separate web pages, to single page applications with client-side rendering and web service back-ends. One of the strengths of web programming technologies is separation of concerns. The concerns such as describing content, styling, behavior, and persistence, are often separated with their own domain-specific languages. A particular programming problem that newer web programming languages tackle is dynamic user interfaces with automatic fine-grained updates. This problem is not restricted to web applications, but applies to any GUI programming abstraction. Consequently, the technologies for web programming are also relevant for development of cross-platform mobile and desktop applications. In this course, students will analyze web languages and frameworks from a programming language perspective. They will explore the underlying concepts and abstractions, and learn how the tools relate to each other. The investigated web technologies range from more traditional full-stack web development solutions with persistence and templating, to popular client-side UI solutions with fine-grained updates and state synchronization.	
Study Goals	The course gives students the conceptual and technical skills to understand the role of programming languages in web programming and the advantages and disadvantages of different approaches.	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab sessions - reading lecture material and papers - making project assignments	
Assessment	Students get a grade for each of the project assignments. The final grade is the weighted average of the grades for the project assignments. There will not be a resit for the course. Disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is the average of the grades for the project assignments.	

CS4400	Deep Reinforcement Learning	5
Responsible Instructor	Dr. J.W. Böhmer	
Instructor	Prof.dr. M.T.J. Spaan	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students must have passed IN4010(-12) "Artificial Intelligence Techniques", or have acquired equivalent knowledge about: - basic probability theory, analysis and algebra - general machine learning methodology, e.g. regression - fully and partially observable Markov decision processes - tabular reinforcement learning methods, e.g. Q-learning - the exploration/exploitation trade-off, e.g. RMAX or UCB - multi-agent learning, e.g. centralized training and decentralized execution Students are encouraged to close any gaps in the above knowledge and to familiarize themselves with the Python/PyTorch deep-learning framework before the start of the course.	
Course Contents	This course will cover the breadth of modern model-free RL methods, discuss their limitations and introduce a variety of current research topics. In particular, we expect to cover the following: - deep learning methodology and architectures - stabilization of approximated value estimation - modern actor-critic methods - planning as inference - exploration with deep networks - offline reinforcement learning - deep multi-agent reinforcement learning - multi-task and meta learning	
Study Goals	After successful completion of this course, students - can list the strengths and limitations of modern deep RL approaches, - explain the underlying concepts of the discussed methods, and how they differ from each other, - can implement selected algorithms/architectures, and - can analyze a new task to decide which algorithms/architectures to apply.	
Education Method	The course will be taught in one lecture per week and the content will be solidified in homework, which will be presented in one mandatory tutorial per week.	
Assessment	The final grade will be 100% determined by a written exam at the end of Q3, with a resit possibility in Q4. To be eligible for the exam, students must attend weekly tutorials and hand in homework exercises. Homework will not be individually graded, but at least 75% of the answers must be of sufficient quality (in terms of time commitment, not necessarily correctness) to be eligible to take the exam.	
Maximum number of participants	As this is the first time the course will be taught, it will be restricted to 30 participants.	

CS4405	Analysis of Concurrent and Distributed Programs	5

Responsible Instructor Dr. B. Özkan

Responsible Instructor Dr. S.S. Chakraborty

Contact Hours / Week x/x/x/x 0/0/4/0

Education Period 3

Start Education 3

Exam Period 3
4

Course Language English

Course Contents

Software systems are becoming highly concurrent and distributed to utilize modern multicore architectures and increasing speed and bandwidth in networks. Shared-memory concurrency in multicore programs and message-passing concurrency in distributed programs share many common abstractions and problems.

In the multicore era, all performance-critical software employs some form of concurrent programming; typically shared memory concurrency. In this setting,
programmers use a number of primitives to develop efficient and correct concurrent programs. To do so the programmers have to understand the behaviors of the primitives and reason about them. It is also important to match the programming paradigms and underlying architectures. For instance, traditionally programmers have assumed that a multithreaded program executed simply by interleaving the executions of its threadsa model known as sequential consistency (SC). This assumption is, however, invalidated both by mainstream multicore architectures, which often execute instructions out of order, and by compilers, whose optimizations affect the outcomes of concurrent programs. As a result, concurrent programs have more outcomes than SC allows.

In the distributed setting, the units of concurrency are independent processes that do not share memory but communicate by exchanging asynchronous messages. The execution of such a system involves two main sources of nondeterminism: concurrency and partial failures. As the processes run concurrently, the exchanged messages can be delivered and processed in many different orderings. The distributed set of processes is also prone to network of process failures. The trade-off between the systems availability in the existence of failures and the consistency between the processes gives rise to a spectrum of weak consistency notions. It is important to reason about concurrency, possible failures, and consistency guarantees to implement distributed programs correctly and understand their behavior.

This course aims to explore analysis techniques for concurrent and distributed programs.

Outline of Lectures:

Shared memory concurrency:
- Abstractions for shared memory concurrency
- Relaxed memory concurrency
- Correctness of concurrent programs

Distributed concurrency:
- Distributed system components, models and assumptions
- Fundamental abstractions for distributed systems

Study Goals

This course aims to give students a deep understanding of concurrency and distribution in modern systems and hands-on experience for analyzing these systems.
At the end of the course, the students will be able to:

- Analyze and reason about concurrent and distributed programs
- Apply and analyze existing techniques on unseen problems
- Be able to pursue independent further research in the area

Education Method

The course consists of the following education methods:

- Lectures for reviewing concurrency and distribution concepts
- Homeworks/assignments
- Developing a course project, writing a report, and presenting it (course project)

To finish the course, students (in teams) will have to:

- Study several papers which will be discussed during the lectures
- Deliver their assignments
- Deliver and present their implementation project

Assessment

The final grade is composed of: research project implementation) (40%) + research project report (20%) + research project presentation (20%) + homework assignment (10% + 10%).
No written exam. Resits are not offered.

CS4415	Sustainable Software Engineering	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	L. Miranda da Cruz	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	Sustainable Software Engineering is an overarching discipline that addresses the long-term consequences of designing, building, and releasing a software project. By definition, sustainability covers five main perspectives: environmental, social, individual, economic, technical. This course mainly focuses on the first, also known as Green Software Engineering. Incidentally, we will also cover some fundamental aspects of social and individual sustainability of software projects. Software Engineering (SE) has long addressed sustainability by narrowing it down to economic and technical sustainability. However, our society is facing major sustainability challenges that can no longer be overlooked by software engineers and computer scientists. It was estimated that, by 2040, the ICT sector will contribute to 14% of the global carbon footprint. Hence, environmental, social, and individual ought to be part of the equation when it comes to design, build, and release software systems. The problem is far from simple, but we need expert computer scientists to bring sustainability into the core values of the next generation of tech-leading organizations.	
Study Goals	After attending this course, you will be able to: LO1. Measure software energy consumption. LO2. Automate carbon-awareness in software development. LO3. Discuss sustainability principles. LO4. Solve sustainability issues in real software projects. LO5. Propose innovative strategies to monitor software sustainability.	
Education Method	To meet these objectives, you will be involved in a broad set of learning activities: lectures, paper reading, software analysis, software development, essay writing and presentation. These heterogenous set of activities aims at building a strong set of hard skills for energy-efficient code development combined with a strong set of soft-skills and critical thinking. Ideally, you will work on projects that will also help real-world software projects embrace a green software culture.	
Assessment	The assessment will be performed as part of the group project. It will include several steering meetings, an essay, a software repository, and a final presentation.	

CS4430	Network Security	5
Responsible Instructor	Dr.ing. A. Zarras	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	The course provides an overview of the most important concepts, methods, and best practices in computer and network security. In this course, students will obtain the knowledge and hands-on experience to secure networking and communication systems. The course's primary focus will be on technologies, protocols, attacks, and defenses. More precisely, starting from a review of common vulnerabilities and attack scenarios, the course will discuss the fundamentals of security engineering and their application in system design, review tools and methods to assess and test communication infrastructure from a security perspective. As a result, students will gain theoretical knowledge and hands-on experience in network attacks and defense methods. Knowledge activation and the transfer from conceptual understanding towards practical experience will be further facilitated by students implementing their own attack or defense tools on selected topics, as well as conducting measurements on the effectiveness of attack and defense schemes.	
Study Goals	See course contents.	
Education Method	Lectures, Labs, and Project.	
Assessment	Assignments and Project.	

EE4560	Information Theory	5
Responsible Instructor	Dr. J.A. Martinez Castaneda	
Instructor	G. Joseph	
Instructor	Dr.ir. J.H. Weber	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	

Course Contents

This course explains the basic ideas of information theory and the correspondences between the elements of this theory and certain natural concepts of importance in a wide number of fields, such as transmission, storage, authoring and protection of data. On the basis of simple concepts from probabiliby calculus, models are developed for a discrete information source and a discrete communication channel. Further, the theoretical basics for developing source coding algorithms is provided, as well as the basics of optimal data transmission through a discrete communication channel.

The following topics will be covered:
* (Differential) Entropy, Relative Entropy and Mutual Information
* Asymptotic Equipartition Property
* Data Compression
* Channel Capacity
* Gaussian Channel
* Rate-Distortion Theory
* Network Information Theory

Study Goals

Upon completion of this course the student will understand the fundamentals of Information Theory, which includes the following: (a) the correspondences between the elements of this theory and certain natural concepts of importance in a wide number of fields, such as transmission, storage, authoring and protection of data, (b) core theorems of information theory, (c) the models that are developed for a discrete information source and a discrete communication channel on the basis of simple concepts from probability calculus, (d) how to develop source coding algorithms, and (e) how to secure optimal data transmission through a (noisy) discrete communication channel.

Education Method

lectures + mini project

Assessment

CoVid-19 disclaimer: In light of the Corona crisis a remote assessment format could be implemented.

Examination: Project and Exam

The grade is determined by a project score (20%) and an exam score (80%). There are two opportunities for both. These are further explained below. Please note the exam format will depend on current CoViD-19 regulations.

Project: The project is individual. Detailed instructions are listed in Brightspace, the project report is to be delivered via Brightspace.

Exam:

If regulation allows standard written examination on campus will be applied, otherwise we will use YouSeeU (Virtual Classroom) which is embedded in Brightspace. All details regarding the examination are listed in Brightspace.

Grading:

First opportunity: The project report should be submitted before the deadline (listed in Brightspace). The project score P1 is an integer between 0 and 10, while the exam score E1 is between 1 and 10 with a half-integer accuracy. The total weighted score is then (4×E1+P1)/5, which is rounded to the nearest grade in the set {5.0, 5.5, 6.0, 6.5, , 9.0, 9.5, 10.0} if both E1 and P1 are at least equal to 5. In case one or both are below 5, then the total weighted score is min(5.5,(4×E1+P1)/5), which is rounded to the nearest grade in the set {1.0, 1.5, 2.0, 2.5, , 5.0, 5.5}. In other words, a necessary condition to pass the course is that both the project score and the exam score must be at least equal to five.
N.B.1: If the project report is not sent before the deadline, then P1=0.
N.B.2: If the student does not participate in the exam, then E1=0.
N.B.3: If the student already did an ee4560 project in a previous study year, then the student can request one of the lecturers before the project deadline by e-mail to let this be taken into account; this can be done in two ways: either to let the score of that project count as P1, or to take a different project from the list and to submit the report before the deadline, in which case P1 will be the maximum of the old and the new score.

Second opportunity: A student not passing in the first opportunity or willing to improve his/her grade can redo the project, redo the exam, or both. In case a new project is done, the topic should be different. The project report should be submitted before the resit deadline (listed on Brightspace). A resit exam will be announced in Brightspace. With the project and/or exam scores being P2 and E2, the new total weighted score becomes (max{P1,P2}+4×max{E1,E2})/5, which is rounded as indicated above, still requiring that both the project score max{P1,P2}and the exam score max{E1,E2} must be at least equal to five.

disclaimer: information may change depending on the developments around the coronavirus.

ET4394	Wireless IoT and Local Area Networks	5
Responsible Instructor	Dr. P. Pawelczak	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Students are advised to follow the course Wireless Communications (ET4358) before taking this Wireless Networking course. An advantage is to have entry-level programming skills (Matlab, Python, C/C++). Nonetheless, students with little knowledge of programming will be helped.	
Course Contents	DISCLAIMER: this study guide information may change depending on the developments around the corona virus. The following modules will be discussed during the lectures: Introduction (example topics): - What is wireless networking - Where to search for (academic) wireless network literature and resources Medium Access Control (example topics): - WiFi: hidden/exposed terminal problem, Carrier Sense Multiple Access - Bluetooth standard: in-depth look into the channel hopping, protocol specifications WiFi (example topics): - Review of IEEE 802.11 standards - Protocol format - ISM band regulation - Adaptive Modulation and Coding - WiFi Matlab class (assignment) IoT networking standards (example topics): - LoRa: protocol specifications, energy consumption, modulation format, network design Review of wireless tools (example topics): - Introduction to wireless packet sniffing and analysis using Wireshark (assignment) - Simple simulations of WiFi network with NS3 RFID networking (example topics): - Principles of backscatter - Protocol formats: EPC C1G2 - RFID hackathon (assignment) Cognitive radio (example topics): - Basics of spectrum management - White Space Databases - Theory of spectrum sensing	
Study Goals	At the end of the course students will be able to: (i) to understand how practical wireless systems work and get a deeper understanding of how the theoretical concepts of wireless communications apply to practice; (ii) employ their own analysis methodology to assess new wireless network systems (especially at the physical layer); (iii) understand rapid prototyping of new wireless systems (for instance, with software defined radio).	
Education Method	Lecture presentations, mini-project assignments, assigned paper reading and its critical analysis and presentation.	
Computer Use	Each student should have its own laptop (preferably with a Linux distribution, where Linux must not be installed on a virtual machine). We will be using Matlab, and/or NS3 and/or GNURadio and/or Wireshark for the assignments.	
Books	WiFi Matlab WLAN toolbox: https://nl.mathworks.com/help/wlan/; Wireshark learn page: https://www.wireshark.org/#learnWS; tutorial on NS3 network simulator: https://www.nsnam.org/documentation/; specific chapters from books provided at the beginning of each lecture.	
Prerequisites	Background in programming (Matlab, Python, Bash)	
Assessment	Points from the mini-project assignments. A research paper analysis from conferences such as IEEE INFOCOM, ACM MobiCom, ACM SIGCOMM will be required to pass the course. disclaimer: information may change depending on the developments around the coronavirus.	

IN4152	3D Computer Graphics and Animation	5
Responsible Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. R. Marroquim	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students that haven't followed any previous Computer Graphics courses (like TI1806) will be able to participate, but might have to invest some more time to catch up in the first lectures.	
Course Contents	Have you ever wondered how Toy Story was made, why the game Last of Us 2 looks so beautiful, or have you ever wanted to create your own graphics application or game? Then you should consider following this course. If not, then you should still follow it... maybe, you will become interested! In this course, you will get a good idea of Computer Graphics in general. The topic is of very high relevance for the industry and the research community and has numerous applications in different domains, such as scientific visualization, video games, simulators, special effects, animated movies and many more. Here, you will learn about basic algorithms, as well as modern techniques. We will address several topics: the principles of image synthesis, object representations, geometric and hierarchical transformations, graphics cards and the graphics pipeline, realistic rendering (including global illumination and effects, such as reflections), expressive rendering, physics simulations, rendering control (including previsualization systems used by professionals in the movie industry), and perceptual rendering, which relies on properties of the human visual system to enhance the quality of the images. Besides course sessions on the theory of Computer Graphics, some of the algorithms will also be reproduced in practice, and deepened during the final project.	
Study Goals	The course teaches computer graphics techniques on an advanced level. After the course the student is able to classify the different modeling, shading, and display techniques. The student can reproduce the basic mathematical and algorithmic notions associated with these concepts, can comment on the weak and strong points of these techniques, and can apply the core concepts within a graphics program in practice.	
Education Method	lectures, instructions, research papers, lab work	
Literature and Study Materials	Research Papers in domain of selected topics, lecture sheets, online sources, optional books (see below)	
Books	Fundamentals of Computer Graphics by Shirley et al. - CRC Press Real-time Rendering by Tomas Akenine-Möller, Eric Haines, Naty Hoffman - Peters, Wellesley Real-Time Shadows by Elmar Eisemann, Michael Schwarz, Ulf Assarsson, Michael Wimmer - Taylor & Francis Computer Graphics. Principles and Practice by James D. Foley, Andries VanDam, Steven K. Feiner - Addison Wesley	
Assessment	The course will be evaluated with two grades, a project grade, accounting for 60%, and a paper grade 40%. The project grade is the result of a project and its presentation that is building upon the assignments that are handed out (roughly) weekly during the duration of this course. The paper grade is the result of the presentation of a scientific paper and the development of an associated practical implementation. Details of both elements will be presented during the lecture. Both grades (project and paper) have to be at least a 5.0 and their weighted average should be 6.0 or higher after rounding (0.5 steps).	

IN4253ET	"Hacking Lab"-Applied Security Analysis	5
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Necessary background differs per student project, see first lecture or contact instructors for details	
Course Contents	The security of computer and telecommunication systems is becoming an increasing concern. In this course, we will review the current state of the art on security research and gain practical experience in assessing the security and vulnerabilities of communication systems. Engineers are typically taught to focus on performance, correctness, scalability, and maintainability when building communication and information processing systems. However, an additional set of design principles are required to achieve security. In this course, we discuss security principles, common pitfalls and vulnerabilities. The weekly lectures provide an introduction into security research, with a focus on real-world security, privacy-enhancing technology and common security pitfalls. Each student participates in a "Hack Project", with a group of one to four students. Students can select between a wide range of available Hack Project outlines within the first week. The goal may be to evaluate the security of a real-world IT system, developing a proof-of-concept exposing a vulnerability or focussed on preserving privacy in a post-Snowden world. Students may propose their own Hack Project based on their background knowledge and skills. Such Hack Projects need to be approved and shaped together with the instructor. Example of possible outlined hardware-oriented projects are: development of a wifi tracker, programing an FPGA system to break passwords, assess the security of RFID cards, or to transparently intercept Ethernet traffic. Concrete software projects are: hacking Bitcoin, improving the TOR anonymity protocol and create Android-based tools for human rights activists in Iran, Egypt and Russia, reprogramming neural networks attacks. Each Hack Project is documented with a written report. This can be in the form of a 6-8 page IEEE-style scientific article or a traditional more lengthy report. All results, experiences and findings are presented to the entire class in the last week of the course. Hack Projects also report their progress several times during the course, after the weekly lectures.	
Study Goals	After this course, the student will have a thorough knowledge of security in real-world systems, and will be able to explore the literature on this topic independently. The student will be aware of the poor state of security in real-world computer systems. The student can explain the common pitfalls, why these known failures still occur and reasons behind the poor state of security in general.	
Education Method	Lectures, student presentations, written final report and active participation. Attendance and active participation during lectures is mandatory. This sadly means telelecturing is not possible.	
Literature and Study Materials	Customize literature lists and study materials are provided per project topic	
Assessment	The final class grade is composed of several partial grades. Partial grades are given for the written Hack Project report (60%), final presentation of result (10%), presentation of ongoing project progress (20%), participation in discussions, overall quality of the practical work and class attendance (10%). Students are required to obtain a passing grade on all partial grades. Attendance to lectures is mandatory. No final written exam. No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	If there is an unexpected high demand for this course, then enrollment will be based on past performance in relevant courses.	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4325	Information Retrieval	5
Responsible Instructor	Dr. C. Hauff	
Responsible Instructor	J. Yang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.	
Course Contents	Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs. Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines. Covered topics include: = Information Retrieval Models; = Indexing Techniques; = Web Search; = Information Seeking Paradigms; = Evaluation of information retrieval systems; = Components of Natural Language Processing; = Word Embedding Techniques; = Neural Language Models; = Deep Contextual Embeddings; = Neural Information Retrieval	
Study Goals	At the completion of this course, students will be able to: = Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1] = Describe and implement different indexing techniques. [Learning Objective 2] = Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3] = Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4] = Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5] = Describe typical NLP tasks and components of an NLP system. [Learning Objective 6] = Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7] = Compare word embedding and contextual embedding techniques. [Learning Objective 8] = Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]	
Education Method	Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments. Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.	
Literature and Study Materials	Scientific papers, course slides, course books - all resources are available on Brightspace.	
Books	Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA. Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press. Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson. Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing. Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".	
Assessment	Weekly individual assignment, weighting 10% of the final grade. Two group projects for IR and NLP, each weighting 45% of the final grade. The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities. disclaimer: information may change depending on the developments around the coronavirus.	

IN4343	Real-time Systems	5
Responsible Instructor	Dr. G. Iosifidis	
Instructor	Prof.dr. K.G. Langendoen	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures & 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	3TU MSc Embedded Systems; the corresponding courses are 2IN26 at TU Eindhoven, and 312030 at TU Twente	
Expected prior knowledge	Basic software engineering, C system programming, basic Linux operating system knowledge	
Course Contents	- basic concepts of RTS - worst case execution time estimation - scheduling policies - response-time analysis - jitter analysis - handling overload - multiprocessor scheduling - reservation-based scheduling	
Study Goals	The course intends to bring the student into the position to: - Explain the fundamental concepts and terminology of real-time systems - Construct task schedules using different scheduling policies under a given set of realistic system constraints - Analyze the timing behavior of a system for a given system model and scheduling policy - Discuss advantages and disadvantages of different scheduling policies for a given platform or system - Discuss the effect of hardware and software interferences on the timing behavior of a given system - Identify (reverse engineer) parameters of a scheduling scheme or a task set from output traces of the system - Derive (reverse engineer) the system specification from a given implementation (in the lab) - Evaluate the scheduling overheads of a given implementation (in the lab) - Implement event-based scheduling policies on a given microcontroller (in the lab)	
Education Method	lectures with exercises (32 hrs); self study (78 hrs); lab assignments (30 hrs)	
Books	Hard Real-Time Computing Systems by G.C. Buttazzo, Springer 2011	
Assessment	Written exam (grade) + lab work; the exam has a resit disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	3	
Permitted Materials during Tests	Simple calculator	

IN4391	Distributed Systems	5
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Starting with the mid-1990s, computing is undergoing a revolution, in which collections of independent computers appear to users as a single, albeit distributed, computing system. Motivated by the increase in the computation capacity of consumer computers, by the commoditization of server-grade machines, and by the advent of the Internet, the distributed computing paradigm has permeated all fields using computers. Current distributed computing applications range from the consumer social networks, peer-to-peer file-sharing, and massively multiplayer online games; to scientific computing using Big Data and distributed sensors; and to engineering fields and industrial control systems. This course focuses on the systems aspects of distributed computing. Specific, contemporary distributed systems are used as illustrative examples to discuss system design and non-functional requirements.	
Study Goals	1. Explain the objectives and functions of distributed computing systems. 2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems. 3. Describe the architecture and operation of distributed computing systems. 4. Explain how distributed computing systems can process user workloads. 5. Explain how distributed computing systems can detect and correct faults and errors. 6. Implement complex operations of modern distributed computing systems in realistic scenarios. 7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)	
Education Method	Lectures: This class uses a partially flipped classroom setting. Students are required to prepare parts of the material through self-study to be able to follow the in-class discussion. This involves, e.g., reading scientific papers. Practical: Designing, implementing, and evaluating a complete distributed system in groups, based on existing research work. Multiple topics are given to choose from. Deliverables include the code and a report of max. 10 pages.	
Literature and Study Materials	Textbook: Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Principles and Paradigms (2nd Edition), Prentice Hall, 2006. The textbook introduces the student to the traditional theory of distributed systems. Additional material: Several relevant research articles introduce the student to the classic literature as well as the latest advances on the topic.	
Assessment	Written exam (closed-book, open questions) during the exam period. A resit for the exam is offered in the following exam period. Practical project assessed based on the code, a presentation, and the report. This course uses gamification. Points can be collected through the practical project (max 4000 points) and the final exam (max 6000 points). The final grade is determined proportional to the 10000 points total. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 4 2021

CS4035	Cyber Data Analytics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	The course provides theoretical and practical background for applying data analytics in the field of cyber security. Cyber data analytics is a huge field with a great diversity of techniques and applications. The course is centered on a selection of seven such techniques: learning from imbalanced data; behavioral profiling and anomaly detection; sequential data mining; data stream mining; learning from software data; adversarial machine learning; and privacy-aware data mining; Anomaly detection is one of the main topics in cyber security. Specific difficulties that the student will learn to handle are the huge amounts of data and the large number of false positives. Behavioral profiling applies to both people and software processes. Different techniques will be taught to construct profiles from software logs. While building such profiles, care should be taken to not infringe upon the privacy of individuals the data is collected from. Finally, attackers will modify their behavior in order to avoid being detected, a cyber data analytics engineer tries to make their models/profiles robust against such modifications.	
Study Goals	The student will be able to: Apply machine learning to real data Understand and modify machine learning algorithms Learn models from time series Detect anomalies in multidimensional time-series Use distributed processing to speed up machine learning Learn models from data streams with limited memory Learn sequential models Use machine learning for fingerprinting and profiling Preserve the privacy of data owners while learning models Learn robust models that can detect evasive attackers Use machine learning to detect fraud, attacks, and botnets	
Education Method	There will be two lectures for each of the seven topics, and 3 large lab assignments on fraud detection, anomaly detection, and behavioral profiling, and 1 smaller lab on adversarial robustness. There is no exam. Teams of two students will work on these assignments which contain both individual and collaborative components. Deadlines are strict as peer-review will be used to both learn of other possible solutions, provide feedback, and get initial estimates on the obtained grade.	
Assessment	3 large lab assignments on fraud detection (30%), anomaly detection (30%), and behavioral profiling (30%), and 1 smaller lab on adversarial robustness (10%).	

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4125	Seminar Research Methodology for Data Science	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Instructor	Dr. K.A. Hildebrandt	
Instructor	J. Urbano Merino	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	basic knowledge in mathematics (linear algebra, calculus, probability and statistics)	
Course Contents	The course focuses on research methods for data science. It looks at underlying principles and concepts for data collection, analysis and data processing, as well as the use of tools to do this. The main topics of study are: Conceptualizing research questions and experimental design Frequentist and Bayesian data analysis Generalized linear models for statistical analysis Multilevel modelling for hierarchical and longitudinal data analysis Measuring and sampling, validity and reliability Linear and nonlinear dimensional reduction Principles of statistical testing In the course, students will be using software tools such as R, and Matlab/Mathematica	
Study Goals	The main aims of this module for the student is to achieve understanding of research methods for data science and obtain practical experience with data analysis and data processing methods. This module provides students with the opportunity to develop and demonstrate their understanding, knowledge, and competence. The learning outcomes for the module are that students will be able to: 1. Appreciate and comprehend strategies for collecting and processing data to answer data-driven research questions 2. Understand and reproduce key principles underlying statistical data and data processing analysis 3. Learn to identify and avoid typical biases, paradoxes and misunderstandings in data-driven research 4. Apply and select appropriate data modelling techniques to analyse data and data processing	
Education Method	Lectures/Assignments Expected Workload Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours (5 × 5 hours for each tool) Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours) Total = 140 hours	
Literature and Study Materials	Will be provided online	
Assessment	Course will be assessed on 3 coursework assignments. A) Analysis of experimental research data (40%) B) Exploration of real-world data set (20%) C) Linear and nonlinear dimensional reduction (40%) Students work in small groups on the 3 assignments. For each assignment, the student group submit a report and give a presentation including a question and answer round where individual group members are assessed on the coursework. The final course mark is the weighted average of the three assignment marks. Note that, there is a minimum grade of 5.0 for each assignment grade and an average grade for all components of at least a 5.8 in order to pass the course. Also, marks for individual assignments do not carry to the next year. Resit next quarter Resubmission of modified coursework is only allowed for assignments that received a fail mark (<5.0). Overall resit mark will be capped to 6.0. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	NA	

CS4140ES	Embedded Systems Laboratory	5
Responsible Instructor	Prof.dr. K.G. Langendoen	
Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/4 Lectures + 0/0/0/4 Lab	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	MUST have C programming skills. Students who have taken the CSE2425 Emb. Software course automatically qualify, others will have to pass an on-line ACCEPTANCE test.	
Course Contents	This highly multi-disciplinary course comes with a lab project where teams of 4 students each will have to develop an embedded control unit for a tethered electrical model quad rotor aerial vehicle (the Quadrupel drone), in order to provide stabilization such that it can hover and (ideally!) fly, with only limited user control (one joystick). The control algorithm (which is given) must be mapped onto a home-brew PCB holding a modern RF SoC interfacing a sensor module and the motor controllers. The students will be exposed to simple physics, signal processing, sensors (gyros, accelerometers), actuators (motors, servos), basic control principles, and, of course, embedded software (C) which is the programming language to be used in order to develop the control system. The project work (including written report) covers the entire duration of the course period, and will take approximately 128 hours, of which 32 hours are spent at the lab facilities. This is a core course of the Masters in Embedded Systems.	
Study Goals	Student is acquainted with real-time programming in an embedded context, along with a basic understanding of embedded systems, real-time communication, sensor data processing, actuator control, control theory, and simulation. Moreover, the student has had exposure to integrating the various multidisciplinary aspects at the system level.	
Education Method	Lectures (8*2hrs), lab work (8*4hrs), coding@home (8*12hrs), report (8hrs), so on average 2 days per week	
Literature and Study Materials	Lecture notes + Website	
Assessment	Lab. project (75%) + written report (25%), no exam, no resit disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	The capacity is limited and -as this is a compulsory course for ES students- they get preference over other MSc students.	

CS4145	Crowd Computing	5
Responsible Instructor	Prof.dr.ir. A. Bozzon	
Responsible Instructor	U.K. Gadiraju	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Basic knowledge of artificial intelligence and/or human computer interaction is advised. Proficiency in at least one programming language.	
Course Contents	Crowd Computing is an emerging field that sits at the intersection of computer science and data science. Crowd computing studies how large groups of people can solve complex tasks that are currently beyond the capabilities of artificial intelligence algorithms, and that cannot be solved by a single person alone. It involves the algorithmic engagement and coordination of people by means of Web-enabled platforms. These complex tasks are mainly focused on the creation, enrichment, and interpretation of data, making crowd computing a building block of data science. Examples of such tasks include the coordinated creation of data about real world events when electronic sensors are not available; the annotation of existing data sets to create ground truth data for the training of machine learning algorithms; and the analysis and interpretation of Web data to spot identify inappropriate content (e.g.,hate speech, or fake news). Crowd computing is an essential tool for any data-driven company: from Facebook to Microsoft, from Google to IBM, from Spotify to Pandora, all major companies employ crowd computing to fulfil their data needs, both by involving employees, and by reaching out to anonymous crowds through online marketplaces like Amazon Mechanical Turk or Appen. Crowd computing methods therefore play an important role in the design, development and evaluation of a variety of products, services, and systems in a variety of domains. The objective of the Crowd Computing course is to introduce the scientific and technical underpinnings of crowd computing, and to investigate how it can be used for computer science applications (e.g., information retrieval, machine learning, next-generation interfaces, and data mining) and for real world applications (e.g., cultural heritage preservation, online knowledge creation, smart cities, etc.) The course is designed around one key challenge, the creation and consumption of (high quality) data, and will be organized around three themes: 1) Establishing data needs; 2) Fulfilling data needs with crowd computing; and 3) Evaluating the quality of the retrieved data with respect to the original data need. Covered topics include: 1) Establishing Data Needs: - Requirement Elicitation - Requirement Analysis - User Modelling Properties 2) Fulfilling Data Needs with Crowd Computation: - Systems for/with collective intelligence (e.g., recommendation, semiautonomous systems, citizen science, crowdsourcing, and human computation systems) - Multi-modal Interaction (e.g., conversational systems) - Human Computation (e.g., worker modelling, task modelling, incentives, task assignment, recruitment) - Games with a purpose - Algorithms for Crowd Computing - Computational Methods for User Modelling - Interfaces for Crowd Computing Systems 3) Evaluating Retrieved Data: - Expert Evaluation - User Evaluation - Explanation of the output of Crowd Computing Systems 4) Study of Application Domains When applicable, the course will also feature invited lectures from selected academics and professionals in the field. Since instructors of this course are also directing the Design@Scale Delft AI lab, students of this course will have the opportunity to engage with cutting-edge research projects relevant to this lab. This Crowd Computing course is an elective for students following the Data Science and Technology Track and the Software Technology Track. It adds to the master education offer by addressing topics that are complementary to courses like IN4325 Information Retrieval, IN4252 Web Science & Engineering, CS4065 Multimedia Search and Recommendation, and IN4010 Artificial Intelligence Techniques.	
Study Goals	After this course, students will be able to: - Identify the requirements for a Crowd Computing system [LO1] - Design and develop Crowd Computing systems. Support and defend the relevance and correctness of his/her choices [LO2] - Describe and compare several Crowd Computing techniques. [LO3] - Describe and compare design decisions in the context of Crowd Computing interaction paradigms [LO4] - Determine which Crowd Computing technique(s) is most appropriate for being used in a certain problem domain [LO5] - Apply the appropriate Crowd Computing technique to an application domain and evaluate the obtained results. [LO6] - Analyse the performance of a Crowd Computing system by applying the proper evaluation measures. [LO7]	
Education Method	** NB: study guide information may change depending on the developments around the coronavirus. This course consists of 16 2-hour lectures. Each week, a 30-minute assignment tests the knowledge acquired on the discussed topics. Starting from Week 1, students form groups and work on a project, to be presented in week 9. Students are expected to work 6 hours per week (each) on the project assignment. Expected workload is 32 hours for attending lectures, 24 hours of reading study material and preparing lectures, 55 hours for weekly assignments and group assignment, 24 hours for preparing final survey, and 5 hours for exam and plenary presentations (total 140 hours).	
Literature and Study	Books:	

Materials	- Human Computation. Author(s): Edith Law and Luis von Ahn. Synthesis Lectures on Artificial Intelligence and Machine Learning, June 2011, Vol. 5, No. 3. http://www.morganclaypool.com/doi/abs/10.2200/S00371ED1V01Y201107AIM013 - A. Marcus and A. Parameswaran. Crowdsourced Data Management: Industry and Academic Perspectives. Foundations and TrendsR in Databases, vol. 6, no. 1-2, pp. 1161, 2013. DOI: 10.1561/1900000044. https://people.eecs.berkeley.edu/~adityagp/papers/crowd-book.pdf - An Introduction to Hybrid Human-Machine Information Systems. Demartini, G., Difallah, D.E., Gadiraju, U. and Catasta, M., 2017. Foundations and Trends in Web Science, 7(1), pp.1-87. https://edu.nl/np4th Slides: available on Brightspace Articles: available on Brightspace Recommended reading: - Interaction Design: Beyond Human-Computer Interaction (4th Ed, 2015). Authors: Jenny Preece, Helen Sharp, Yvonne Rogers
Assessment	The final grade consists of the following parts: - Weekly Individual assignment, weighting 15% of the final grade - Group assignment, weighting 55% of the final grade - Final Individual Assignment (Survey), weighting 30% of the final grade The group assignment is performed collectively, but graded individually. Assignments have no re-sit opportunities. Disclaimer: information may change depending on the developments around the coronavirus.
Tags	Algorithmics Artificial intelligence Design Programming Software
Co-Instructor	J. Yang

CS4205	Evolutionary Algorithms	5
Responsible Instructor	Prof.dr. P.A.N. Bosman	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Must have the ability to program your own (extensions of) evolutionary algorithms (e.g., C, C++, Java). Further, although no particular high-level prior knowledge is expected, a basic understanding of key aspects in linear algebra (e.g., matrix multiplications, inversions, decompositions, etc.), probability theory and statistics (e.g., normal distribution, statistical hypothesis testing), algorithm design (e.g., efficient local search, heuristics), complexity theory (e.g., big-O notation, algorithm analysis), will be very helpful.	
Course Contents	In this course we consider a specific subfield of Artificial Intelligence: Evolutionary Algorithms (EAs). These algorithms, sometimes also identified as being part of the class of bio-inspired algorithms, have as a metaphor the concept of natural evolution, i.e., the mechanisms by which, the fittest individuals in a population survive, reproduce, and in doing so, over time, change to be better equipped to thrive in their environment. Initiated in the 60s and 70s of the 20th century, research on EAs has progressed immensely. Today, EAs are being used to solve real-world problems in many areas, e.g. to optimize the layout of electrical wind farms, to automatically create radiation therapy treatment plans, and to optimize the architectures of deep neural networks. This course covers a spectrum of topics in EAs, ranging from basic concepts to advanced, recent, and state-of-the-art research, and ranging from theoretical to applied. In particular, topics include genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, optimal mixing evolutionary algorithms, multi-objective optimization, and real-world applications. The course is planned to have 7 lectures and 2 practical assignments. The first practical assignment pertains to experimenting with already implemented EAs on predefined problems. The second practical assignment offers more freedom, allowing you, in a group, to build your own EA (this may vary depending on student numbers and other circumstances). disclaimer: information may change depending on the developments around the coronavirus.	
Study Goals	Upon successful completion of this course, students will be able to: 1) Explain the key concepts underlying the main streams in Evolutionary Algorithm (EA) research, with in particular genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, and optimal mixing evolutionary algorithms. 2) Explain key ingredients underlying the rationale of when these algorithms work and when they do not work. In particular: schema analysis and how the match between the search bias of an EA and the fitness landscape is influenced by aspects such as variable dependencies and multi-modality. 3) Name and explain key research lines along which state-of-the-art research in EAs is done to achieve more robust, efficient, and effective EAs. 4) Identify good opportunities for using EAs, or hybrid versions thereof, in practice. 5) Properly (scientifically) experiment with EAs as well as program your own.	
Education Method	7 Lectures 2 Lab projects disclaimer: information may change depending on the developments around the coronavirus.	
Literature and Study Materials	Papers and slides that will be made available.	
Assessment	The final grade is based on 60% written exam, 40% lab practical work. disclaimer: information may change depending on the developments around the coronavirus. In particular, there may be no written exam. In that case, there will likely be several smaller practical assignments and a large practical assignment at the end of the course. Both parts will then likely count for 50%. For the large assignment there will also be chance to resit this part by means of a repair option through an oral examination of the lecture contents.	
Permitted Materials during Tests	None	
Tags	Algorithmics Artificial intelligence Optimalisation	

CS4210-B	Intelligent Decision Making Project	5
Responsible Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Dr. J.W. Böhmer	
Instructor	Dr. E. Demirovi	
Instructor	Dr. N. Yorke-Smith	
Instructor	Prof.dr. M.M. de Weerdt	
Contact Hours / Week x/x/x/x	0/0/0/1	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Theoretical knowledge regarding algorithms for decision making in Artificial Intelligence, obtained for instance by passing one of the following courses: - CS4210-A Algorithms for Intelligent Decision Making - CS4400 Deep Reinforcement Learning - IN4010(-12) Artificial Intelligence Techniques - IN4344 Advanced Algorithms.	
Course Contents	Decision making is at the centre of artificial intelligence. Building upon theoretical knowledge gained in other courses, students collaborate in small groups on a distinct research project per group, for instance on decision-making problems in transport, logistics or smart energy grids. Purely algorithmic challenges will also be provided. The research projects provide a good opportunity to learn about topics suitable for Masters projects in the Algorithmics section.	
Study Goals	After completing the Intelligent Decision Making Project course, the student is able to: 1. Apply algorithms for decision making to problem domains, and can compare and evaluate them. 2. Design and implement an extension of a decision-making algorithm. 3. Identify and discuss relevant topics in the research field of algorithms for intelligent decision making. 4. Describe and apply the appropriate research methodology. 5. Communicate his/her findings effectively.	
Education Method	A research project in a small group.	
Literature and Study Materials	Mainly survey papers and book chapters. Details are provided via Brightspace.	
Assessment	The assessment consists of the following items: 1. Quality of work of the research project (40%) 2. A scientific report of the research project (including peer review of a report) (20%) 3. Performance during the project (30%) 4. Oral presentation of the research project (10%) Only items 1 and 2 can be examined a second time. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Only a limited number of students can participate in this course. In order to be admitted, please submit a short motivation letter (max 200 words) via Brightspace. Attending the first lecture is compulsory.	
Tags	Artificial intelligence	
maximum aantal deelnemers	40	

CS4265	Computer and Network Security: Advanced Topics	5
Responsible Instructor	Prof.dr. M. Conti	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	*DISCLAIMER: study guide information may change depending on the developments around the coronavirus.* The course takes the form of seminars based on a selection of scientific papers (that either have had a strong impact on security today, or explore novel ideas that may be important in the future). The list of topics can be found in the brightspace Topics and Papers module. For each topic there is a primary paper, and possibly other additional papers. All the students are required to read all primary papers and be able to competently discuss the material in class. Each student will be responsible for presenting one lecture (based on one of the primary paper including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion in the class. 48 hours before each lecture each student must upload on a shared repository at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). This is intended to be an interactive class: class participation is strongly recommended (and will play a role in the grading criteria). Sleeping during the class is optional, but not recommended.	
Study Goals	This course is about learning to study, analize, do and criticize research in cybersecurity. This will be done by being exposed to actual research topics and scientific papers and discussing things together.	
Education Method	Studying, presenting and discussing recent research results in Computer and Network Security.	
Assessment	Presentation + Class Discussion + Written Report + Oral Exam (please refer to the Judgement field for more details) disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Judgement	The final grade will be made up of four components: 25% the presentation done by the student during the course: each student will be responsible for presenting one topic (based on the corresponding primary paper, including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion (Q&A) in the class. This component is based on following criteria: (15%) Layout and Graphics (30%) Content (20%) Organization (20%) Presentation (15%) Q&A. 25% for the active participation in Q&A sessions during the course: 48 hours before each lecture each student must submit (via email, to both the lecturer and the teaching assistants) at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). The students should actively participate in the discussion of the topics in the 10 minutes Q&A session for each presented topic. 25% for content and quality of the final essay: At the end of the course, each student must write a 5-page long essay about one of the topics that has been discussed in class, or another topic agreed with the lecturer. The topic and the structure of the essay must be agreed with the lecturer. The essay might include some implementation prototype or experiments/simulations to evaluate/support the claim in the paper (in case this is a significant part of the essay, two students can agree with the lecturer to work together). If the student cannot attend the lectures, an alternative work (e.g. a longer essay) must be agreed with the lecturer. 25% for the oral presentation of the essay: during the oral exam, the student is asked to give a 15-minute presentation to the lecturer and the teaching assistants about the essay (presenting with slides is highly recommended). During the oral presentation, students can also be asked questions about other topics of the course. This component is based on following criteria: (30%) Style (20%) Originality (50%) Organization (clarity in your argumentation, coherence between assumptions and conclusions, logical organization, evidence to support claims)	
Co-Instructor	Ir. S.E. Verwer	

CS4280	Language-Based Software Security	5
Responsible Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course has no formal prerequisites. However, for the homework assignments you will have to implement several program analysis techniques using the Scala programming language. If you have not used Scala before, you are thus expected to learn the basics of the language through self-study.	
Course Contents	Security vulnerabilities often arise due to programming errors in the source code of an application. Recent programming errors with severe security implications include Heartbleed (buffer over-read), Shellshock (code injection), and goto-fail (ill-formated code). Rather than hunt for individual vulnerabilities in programs, a more structural approach to improve security is to improve the programming language. This is the goal of language-based security: to rule out whole classes of potential security vulnerabilities in one go. This course studies various security properties and program analysis techniques for enforcing these properties at the level of the programming language to improve software security. In particular, we will study the following properties: - Memory safety: prevent buffer overflows and overreads - Type safety: prevent undefined behaviour - Information flow control: prevent data leaks and code injection attacks We will study techniques to address these problems at the language level through dynamic analysis, static analysis, and language design. To facilitate a precise study and comparison, we will define the above techniques formally in class. To facilitate student experimentation and exploration of trade-offs, students will implement the above techniques in homework assignments.	
Study Goals	After taking this course, students should be able to: 1. Describe the nature and causes of security vulnerabilities in software systems, and give concrete examples of how these security vulnerabilities can be exploited. 2. Explain the properties that can be enforced at the level of the programming language to rule out security vulnerabilities, such as memory safety, type safety, and non-interference. 3. Formally define the semantics of a simple programming language. 4. Formally define dynamic and static analysis techniques for enforcing these security properties. 5. Implement these techniques for a small programming language. 6. Discuss and evaluate the importance of soundness and precision of a given program analysis. 7. Contrast programming languages based on the set of countermeasures they provide, and give an appropriate recommendation for a specific application. 8. Analyse and apply results from scientific literature in the area of language based security.	
Education Method	The course work consists of the following activities: 1 or 2 instruction sessions per week. Weekly homework assignments consisting of theoretical questions, programming assignments, and reading assignments	
Assessment	The assessment for this course consists of two parts: The weekly homework assignments will test your ability to design an implement (variants of) the techniques discussed in the lectures (study goals 3-5). This counts for 40% of the total grade. The final written or oral exam will test your theoretical understanding of the security vulnerabilities and their countermeasures discussed in class (study goals 1-2) and your ability to discuss and contrast the different aspects of these techniques (study goals 6-8). This counts for 60% of the total grade. To pass the course, each of these grades (homework assignments and final exam) should be 5.0 or higher, and the final grade should be 5.8 or higher (and will be rounded to the nearest half grade point). disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	E. Visser	

CS4290	Seminar on Distributed Machine Learning Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	Machine learning systems are often conventionally designed for centralized processing in that they first collect data from distributed sources and then execute algorithms on a single server. Due to the limited scalability of processing large amount of data and the long latency delay, there is a strong demand for a paradigm shift to distributed or decentralized ML systems which execute ML algorithms on multiple and in some cases even geographically dispersed nodes. The aim of this seminar course is to let students learn how to design and build distributed ML systems via paper reading, presentation, discussion, and project prototyping. We provide a broad overview on the design of the state-of-the-art distributed ML systems, with a strong focus on the scalability, resource efficiency, data requirements, and robustness of the solutions. We will present an array of methodologies and techniques that can efficiently scale ML analysis to a large number of distributed nodes against all operation conditions, e.g., system failures and malicious attacks. The specific course topics are listed below. The course materials will be based on a mixture of classic and recently published papers. For each topic, the basic concepts and technology landscape will be first provided and then two state-of-the art of papers will be presented and discussed by students. We offer a testbed of a distributed (deep) ML system in which students are encouraged to apply different techniques to jointly improve its scalability and resilience. Course topics include Overview of distributed machine learning systems Performance and scalability of state-of-the-art systems Acceleration of ML workloads Slim distributed ML systems on small data Robust deep learning systems Federated machine learning systems	
Study Goals	Students are able to argue and reason about distributed ML from a systems perspective. Students understand the behavior and tradeoffs of distributed ML in terms of performance and scalability. Students can estimate the importance of data inputs via different techniques, i.e., core set and decomposition methods, for distributed ML systems. Students understand data poison attacks and design defense strategy for distributed ML systems. Students can analyze the state-of-the art federated machine learning systems and design the failure-resilient communication protocols. Students are able to design and implement methods and techniques for making distributed ML systems more efficient.	
Education Method	Lectures: 7 weeks X 2h Papers: one paper presentation, two paper reviews, and paper discussion. Practical: apply system and algorithmic optimization techniques learned in the lecture to improve the performance of distributed machine learning systems, e.g., image recognition on CIFAR 10. The testbed environment, learning algorithms, and dataset will be given. Deliverables include git commit of functioning code and a report summarizing the contribution	
Assessment	Paper presentation by group (10%): each group of 2 to 3 students needs to choose from a given set a papers to present (15 minutes) and lead the discussion (10 minutes). Paper reviews by individual (30%): each student needs to write three reviews of papers assigned from a given set and those two papers have to be different from the paper for the presented. Each review will account for 10% of the grade. Questionnaires by individual (0%): each student needs to hand in a list of questions at the beginning of the lectures that have paper presentation from other students. This is not graded. Individual project (60%): The project is collaborative among the entire class and competitive as a whole group. The objective is to continuously improve the performance of the given distributed ML system. The students need to hand in a final project report in style of a short scientific paper, stating their individual contribution to the overall system performance. disclaimer: information may change depending on the developments around the coronavirus.	

CS4295	Release Engineering for Machine Learning Applications	5
Responsible Instructor	L. Miranda da Cruz	
Responsible Instructor	Dr.ing. S. Proksch	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	The world of Software Engineering has been revolutionized in the last decade. Instead of releasing software updates yearly, companies can now release multiple times per week, sometimes even per day, to their customers. This allows much quicker reactions to market demands, software failures, and is crucial to increase the business value of software. These improvements have been mostly enabled by advances in release engineering and, in this course, we will learn about the techniques and technologies that build the foundation for modern release engineering. We will go on a journey that starts at continuous integration and then moves on to continuous delivery, continuous deployment, and continuous experimentation. We will discuss the theory and the current research on various related subjects like containerization, testing, or monitoring and will put the learned theory into practice. As a running example, we will build a pipeline for a machine learning application, which -compared to traditional release engineering- poses additional challenges, like data versioning or model deployment.	
Study Goals	After following this course, students are able to... - Apply standard techniques of release engineering - Apply version control techniques to machine learning artifacts, like data or models - Design a deployment pipeline for a machine learning application - Implement quality control techniques in a machine learning pipeline - Analyze and improve existing deployment pipelines - Evaluate and document design decisions in deployment pipelines	
Education Method	- Following interactive lectures - Active participation in tutorial sessions - Reading scientific papers and gray literature - Performing a small literature survey - Implementation of a pre-defined release engineering pipeline - Deriving and implementing an improvement for the pipeline - Documenting the improvement in a scientific essay	
Assessment	Formative Assessment: - Individual group meeting for feedback on current pipeline and pipeline extension proposal - Written feedback on Table of Contents and Introduction of written essay - Individual group meeting for feedback on project progress - Written feedback on methodology and pipeline of written essay Summative Assessment: 35% Final release engineering pipeline (focus: how well is the project executed) 60% Essay (focus: how well have design decisions been evaluated and documented) 5% Presentation (focus: clarification and fraud prevention) Please note: - The different parts of the summative assessment represent grading components and need ALL to be passed to receive a positive overall grade. - There is NO resit opportunity for this course. - Partial grades are not carried over to the next academic year.	
Special Information	The course information presented in the study guide may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. A.E. Zaidman	

CS4410	Category Theory for Programmers	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Categorical structures occur in programming languages on different levels: (1) within programming languages, providing design principles and guidance on how to write modular and correct-by-design programmes (as demonstrated in the practical programming language Haskell) and (2) in the design and study of programming languages, as a guiding meta-theory. In particular, category theory provides a mathematical justification for recursion schemes for inductive datatypes. This course aims to provide solid foundations on both (1) and (2).	
Study Goals	- Use categorical constructions (e.g., monads) in the design and structuring of computer programmes in Haskell - Prove properties of computer programmes, guided by categorical intuition - Understand categorical fusion laws and how to use them to optimize code - Understand the theory of infinite data structures and apply it to practical problems	
Education Method	Learning in this course is achieved through lectures, problem sessions, and guided self-study.	
Assessment	Exam at the end of the term, counts for 100% of the mark.	

EE4715	Array Processing	5
Responsible Instructor	Dr.ir. R.C. Hendriks	
Responsible Instructor	Prof.dr.ir. G.J.T. Leus	
Instructor	Prof.dr.ir. A.J. van der Veen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Linear algebra, signal processing, Fourier transform, stochastic processes and preferably statistical signal processing and some experience with matlab	
Summary	In this course we discuss array processing techniques for signal separation and parameter estimation, using arrays of sensors. After a review/introduction of the necessary linear algebra tools we will start with deriving the signal processing model for narrowband applications, followed by the wideband extension, and apply these to several applications among which array processing for wireless communication, audio and speech processing, biomedical signal processing and astronomy.	
Course Contents	Signal processing models for narrowband and wideband array processing, elementary beamforming concepts (spatial filtering), tools from linear algebra: QR, SVD, eigenvalue decompositions, projections and GEVD. Elementary beamformers/receivers: the matched filter, the Wiener filter, MVDR, LCMV, etc. Estimation of angles and delays using ESPRIT, adaptive space-time filters, the LMS algorithm and factor anaylsis.	
Study Goals	To be able to explain some key problems regarding data models, estimation and detection that occur in array processing applications. - To be able to explain the major signal processing tools required to solve array processing problems. - To be able to implement these signal processing techniques in Matlab. - To be able to apply these techniques to new array processing problems.	
Education Method	Lectures + mini project	
Literature and Study Materials	References from literature and notes	
Assessment	Oral exam: Take-home assignment with oral discussion of the results	

ET4030	Error Correcting Codes	4
Responsible Instructor	Dr.ir. J.H. Weber	
Contact Hours / Week x/x/x/x	0/0/0/3	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	A B.Sc. Programme in Electrical Engineering, Computer Science, or Mathematics	
Course Contents	Introduction into error-correcting codes; mathematical basics; block codes fundamentals; cyclic codes; co-operating codes; soft-decision decoding; convolutional codes; iterative decoding (turbo codes, LDPC codes); applications.	
Study Goals	The global goal of this course is to get acquainted with the basics and applications of error correction coding techniques. Such techniques are applied in order to protect information against errors which may occur during transmission or storage. The specific techniques under consideration in the course are the ones discussed in the lecture notes, which may be updated from year to year according to recent developments. The emphasis will be on the basic trade-offs between efficiency, reliability, and complexity. Unless explicitly indicated, the proofs of the results are not part of the course contents (the interested student may consult books from the bibliography). In the end, the student should be capable of making choices for suitable error correction coding techniques in the context of information transmission and storage applications. The student has to demonstrate to have understood the aforementioned techniques and trade-offs by solving exercises in a closed-book written or oral exam. The level of these exercises is similar to the examples and exercises provided in the lecture notes.	
Education Method	Lectures; expected workload is 22 hours attending lectures, 60 hours preparing for the lectures, studying the lecture notes, and making suggested exercises, and 30 hours for preparing and making the exam.	
Literature and Study Materials	Lecture notes "Error-Correcting Codes" by J.H. Weber	
Assessment	The final grade will be fully determined by a scheduled written exam, which will be held at the end of Q4. If it turns out that this is not possible, then an individual remote oral exam opportunity will be offered instead, also at the end of Q4. The resit will take place as a (remote) oral exam at the end of Q5, on appointment with the lecturer. The exam format is closed-book in any case.	
Remarks	The above-mentioned information may change depending on the developments around the Corona virus. Actual course information available on Brightspace.	

ET4285	Measuring and Simulating the Internet	4
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	(Advanced) Networking course (e.g., CS4055) and Programming skills.	
Course Contents	The Internet is a complex network without a fixed structure. Hence, measuring the Internet is crucial to acquire knowledge about the Internet infrastructure (topology), traffic, and performance (e.g., loss, delay, bandwidth, etc.). This course will discuss the design requirements and challenges in measuring and simulating the Internet, and the existing measurement methodologies (how/where/when to measure). Knowledge of how to conduct and evaluate Internet measurements enables the design and enhancement of a large set of applications, including: capacity planning and traffic engineering, network management and trouble-shooting, detecting network abuse and intrusions, etc.	
Study Goals	The goal of this course is to introduce the students to basic Internet measurement tools, as well as the state-of-the-art in Internet measurements research. The students will learn several Internet measurement techniques (e.g., active vs. passive measurements), and different software tools. Through a measurement assignment, the students will learn how to define/formulate a research problem, choose a specific approach, and complete a measurements-related research project.	
Education Method	Weekly instructions (8x2 hours) + independent project work (8x12 hours).	
Literature and Study Materials	Papers	
Assessment	Groups of students will be assigned a project that requires the students to put the theory on measuring and simulating the Internet into practice. The students have approximately 1 month to complete their assignment. The final assessment is based on the presentation (via report and/or demonstration) of the project assignment results and on the individual contribution and level of participation. Students within a group may thus receive different grades. As this is a project-based course, there is no official resit scheduled. Instead, an opportunity will be given to improve the work.	
Remarks	Disclaimer: The information about ET4285 may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	Because this is a project-based course, we can only admit a limited number of students (typically around 30, but the actual number depends on the number of TAs involved). If more students enrol, we will give preference to those who have successfully completed CS4055.	

IN4185	Globally Distributed Software Engineering	5
Responsible Instructor	Prof.dr.ir. D.M. van Solingen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	Exam by appointment	
Course Language	English	
Expected prior knowledge	Software Engineering (= IN2705)	
Course Contents	The course Globally Distributed Software Engineering (GDSE) will address pro's and con's of GDSE, practical consequences of GDSE, technological (in)feasabilities for GDSE, and practical experiences and examples of GDSE for example in outsourcing, off-shoring, near-shoring and multi-partner systems development. The central theme of this course is the fact that software engineering is carried out in practice more and more in globally distributed settings. This has advantages and disadvantages that need to be addressed in a practical matter when carrying such projects. The course is run asynchronous in BrightSpace. Lectures and excercises are followed digitally in weekly modules that need to be followed prior to the weekly synchronous lecture/virtual meeting. The course hours in the calendar are used for interaction with the professor and more detailed discussion and feedback. The course builds upon individual discipline in preparing the weekly modules online, in combination with group assignments during these weeks as well. Also the group assignments (in groups of 3 or 4 students) can be done virtually.	
Study Goals	The course Globally Distributed Software Engineering (GDSE) aims at teaching participants (1) the technical and organisational setting of carrying out software engineering in practice when distributed over the world, and (2) understanding best-practices in collaboration in software engineering project teams that carry out their work in a distributed setting.	
Education Method	Digital lectures, quizes, group assignments and online discussion. These are used as preparatory work prior to the weekly synchronous lectures (that are merely virtual as well), weekly group home work assignments and individual assignments.	
Computer Use	The course does not contain programming excercies. Though in the group assignment students will have to create a deliverable of choice. This can be very broad from creating a YouTube instruction video to writing an online book, or from creating a Wikipedia page to setting up tooling environment.	
Literature and Study Materials	Presentation handouts	
Assessment	Written report on lab work and literature research, individual f2f examination meeting of 30 minutes with professor. The course grade is calculated from the group assignment (25%), individual essay (25%), personal interview on GDSE course and individual essay (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Please enroll. If enrolled please pay attention that Module 1 of this course needs to be finished PRIOR to the first lecture meeting! Every week a new module is released in BrightSpace that needs to be worked through prior to the weekly synchronous meeting.	
Special Information	Please contact d.m.vansolingen@tudelft.nl	

IN4254	Smart Phone Sensing	5
Responsible Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Requirement 1: Students MUST either (1.1) have passed a JAVA programming course, or (1.2) have passed a C/C++ programming course and be familiar with JAVA, or (1.3) know Objective C (programming language for MACs). This requirement is equivalent to having passed the course TI 1206 in our first year Bachelor curriculum "Object Oriented Programming" Requirement 2: Students MUST (2.1) have passed a basic course on Probability Theory. This requirement is equivalent to having passed the course TI 2216M in our second year Bachelor curriculum "Probability and Statistics". We will be refreshing some concepts on Probability, but we will not be refreshing concepts on Object Oriented Programming.	
Course Contents	The course provides an introduction to the current research trends in the area of smartphones. The course will be based on a programming project, where students will form groups of two to develop a smartphone application. This is not a programming course; students are expected to have already programming experience. To develop a smartphone application, a user needs to be familiar with (1) the signals and data that smartphones can gather, and (2) the mathematical tools necessary to process this data. This course will provide a solid background for the above two points. During the lectures we will analyze the latest research papers on this emerging field. We will dissect these papers to understand how techniques from algorithms, signal processing and machine learning are used to develop some exciting applications. The students will then use these basic technical tools to develop their own apps.	
Study Goals	The goals of this course are twofold. First, to expose students to the increasingly important area of mobile computing. Students will learn how mobile phones can be used to solve problems in areas ranging from health care and indoor localization to song recognition and traffic management. Second, to provide students with a basic set of tools to develop their own applications. For students aiming for industry, the course should enhance their ability to use theoretical tools to solve practical problems. For students involved on research activities, the course will provide them with the necessary background to use smartphones as a distributed sensing and processing unit that could be used to solve the particular problems in their areas. After taking this course students will be able to: (1) Explain the current applications, methods and research trends in the area of smartphone sensing. (2) Apply key mathematical tools in the development of smartphone applications. (3) Analyze how a sensing and computing problem can be solved via the use of smartphones, and identify the steps required to design a solution. (4) Create a non--trivial and innovative smartphone application.	
Education Method	Lectures + Lab The project work, including the written report, covers the entire duration of the course period, and will take approximately 120 hours, of which 14 hours are spent on lectures, 10 hours preparing reports, 10 hours reading research papers, and the remaining part programming the App (the time spent in the Lab belong to this latter part).	
Literature and Study Materials	Research Papers and web tutorials	
Assessment	Written reports + project presentation + oral exam Overall, the final grade is determined by: 1) Two intermediate reports (5% of grade each, 2 pages each) 2) Final report (10 % of grade, 5 pages) 3) Final project demonstration (80% of grade) The first two reports are due on the third and fifth week; and the final report, project and exam are due on the ninth week. There is no resit for this course. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	1. You need to enrol in Brightspace 2. The first lecture will be compulsory 3. This course can only accommodate 60 students, with ES students having a preference when demand exceeds capacity. If your program marks this course as required, you are guaranteed a spot. IMPORTANT: The study guide information may change depending on the developments around the coronavirus.	

IN4255	Geometric Data Processing	5
Responsible Instructor	Dr. K.A. Hildebrandt	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Basic knowledge in mathematics (linear algebra, calculus): TI1106M, TI1206M or comparable courses. Students who haven't followed any of these courses can follow the course, but should be willing to invest more time.	
Course Contents	Geometry processing is concerned with the representation, analysis, manipulation, and optimization of digital shapes. Thanks to the advances in 3D acquisition and manufacturing technologies (like 3D-Scanning and 3D-printing), the usage of geometric data is continuously increasing and an efficient processing of digital shapes plays an important role for a variety of applications in areas such as computer graphics, computer-aided design and engineering, medical imaging and surgery planning, architecture, and entertainment. In this course, we will study concepts and algorithms for creating, analyzing, editing and optimizing digital geometric shapes.	
Study Goals	After successfully completing this course, the student is able to: - describe the fundamental techniques used for representing, analyzing, processing and modeling digital 3D-shapes treated in the course and to explain the mathematical and algorithmic concepts associated with them - apply the learned mathematical concepts to solve basic geometric problems arising in geometric modeling applications - design algorithms that can solve simple geometric modeling tasks and evaluate the drawbacks, benefits and limitations of the proposed algorithms - implement the designed algorithms in a geometric modeling software framework	
Education Method	The course combines lectures, tutorials, practical project work, and homework assignments.	
Literature and Study Materials	References to textbooks and recent research and survey papers are given in the lectures.	
Assessment	The course will be assessed on two practical projects and two theoretical assignments. The course grade is a weighted average of the grades of the practical projects (60%) and the theoretical assignments (40%). Note that, there is a minimum grade of 5.0 for each assignment grade and the average grade for all components of at least a 5.8 in order to pass the course. Also, grades for individual assignments do not carry to the next year. Resubmission of modified coursework is only allowed for assignments that received a fail grade (<5.0). Overall resit grades will be capped to 6.0 disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4331	Web-scale Data Management	5
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Dr. A. Katsifodimos	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Bachelor level courses in database management systems and operating systems. A prior course in distributed systems or middleware would be helpful but is not required. Programming skills are important for the final assignment.	
Course Contents	This course addresses the challenges of Data Management at Web-scale. Especially, it covers the need for large-scale distributed data storage systems. The lecture therefore introduces step-by-step increasingly complex distributed storage systems, leading up to modern implementations of different NoSQL data storage systems. The challenges arising from such systems are presented and discussed, especially focusing on the CAP theorem and the resulting trade-offs with respect to data models, transactional power, query expressivity, and replication consistency. These discussions lead to different variants of NoSQL database systems, like Key-Value Stores, Document Stores, Wide-Columnar stores, and Graph Databases. The advantages, disadvantages, and general properties of these systems are discusses in more detail. There is special focus on distributed transactions and consistency guarantees of different data management systems and methods.	
Study Goals	At the end of this course the student can - assess the nature of a given storage problem, and can select a suitable technology for solving it - understands the different data models encountered in Web Data Management, and their impact on modelling and querying - understands the issues arising from distributing and replicating data, especially with respect to the CAP theorem -understands the trade-offs which can be chosen within the design space of the CAP theorem -categorize and explain modern NoSQL databases within the framework of the previously mentioned trade-offs	
Education Method	Lectures and assignments	
Literature and Study Materials	Course slides and Lecture Videos	
Books	Literature mentioned in the lecture, mostly research papers.	
Assessment	Depending on the year, there are group assignments with final presentation, written report, and oral interviews. Final grade will be the weighted average of the three partial grades (the weights will be announced during the lecture depending on the available topics and assignments in the current quarter). The interview covers questions from the topics of the lecture. The interview can be resit, the assignment & presentation cannot. disclaimer: information may change depending on the developments around the coronavirus.	

IN4333	Language Engineering Project	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/0/0/4 (lab)	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Compiler construction CS4200-A and CS4200-B.	
Course Contents	"Software systems are the engines of modern information society. Our ability to cope with the increasing complexity of software systems is limited by the programming languages we use to build them. Bridging the gap between domain concepts and the implementation of these concepts in a programming language is one of the core challenges of software engineering. Modern programming languages have considerably reduced this gap, but often still require low-level programmatic encodings of domain concepts. Or as Alan Perlis formulated it in one of his famous epigrams: 'A programming language is low level when its programs require attention to the irrelevant'. A fixed set of (Turing Complete) programming constructs is sufficient to express all possible computations, but at the expense of considerable encoding that obfuscates the concepts under consideration. Linguistic abstraction can be used as a tool to capture our emerging understanding of domains of computation." (Visser, SCP 2015) In the precursor compiler construction course (CS4200), students learn the basics of language engineering by building a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts. In the precursor compiler construction course (IN4303), students learn the basics of language engineering and build a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts.	
Study Goals	In this course students learn to apply language engineering principles and tools to a real (domain-specific) programming language. Explore the definition of all aspects of a programming language: syntax, name binding, type analysis, transformations, code generation.	
Education Method	This is a project course. Students deepen their language engineering skills and insights by building a complete language definition. Students work in teams of two on the definition of a (domain-specific) programming language using the Spoofax Language Workbench. Assistance and feedback is provided during weekly lab hours. The project should span the full life cycle of language implementation including a test suite, IDE, code generator, and distribution of the result as an Eclipse plugin.	
Literature and Study Materials	- Documentation of the design and implementation of a specific language - Papers about language definition techniques	
Assessment	The work is assessed based on a code review of the language definition, a written report about the project, and a presentation in the final project workshop. The course has no resit. disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is based on the following components: - git repository with language project (40%) - written report about the project (30%) - presentation (slides) (30%)	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Track Artificial Intelligence Technology 2021

Introduction 1

1. An Individual Exam Programma (IEP) in this track consists of
a. a common core,
b. courses offered by the faculty EEMCS,
c. a seminar offered by the programme CS or a Literature survey (IN4306)
d. free electives,
e. a thesis project (IN5000 Final project) worth 45 credits and
f. if required, homologation.

The IEP must be drawn up in agreement with the thesis coordinator of the research group in which the student wishes to carry out his or her thesis project. The thesis coordinator is a member of the scientific staff of that research group.

the seminar of the research group in which the thesis is performed or the Literature Study (IN4306) is part of said IEP,

Free elective courses
c. the number of credits spent on free electives in said IEP is no higher than 25 credits,
d. the number of credits spent on homologation in said IEP is no higher than 15 credits,
e. at least 40 credits of the courses in the IEP (notwithstanding the thesis project) should be computer Science courses. A list of these courses is published annually in the digital study guide.

Free electives - language course list:
Up to 3 credits may be spent on language courses. These may only be chosen if required. Placement tests showing the necessity to take one or more of these courses must be taken and submitted to the master coordinator.

WM1101TU English for academic purposes-3 3
WM1135TU English for academic purposes-4 3
WM1136TU Written English for technologists-1 3
WM1102TU Written English fortechnologists-2 3
WM1137TU Spoken English for technologists-1 2
WM1112TU Spoken English for technologists-2 2
WM1115TU Elementary 1 Dutch for foreigners 3
WM1116TU Elementary 2 Dutch for foreigners 3
WM11117TU Dutch intermediate 1 3

The free elective space may also be used for an extra project:
TUD4040 Joint Interdisciplinary Project (JIP) 15

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Common Core AIT 2021	
Introduction 1	Common Core MSc CS - DST (at least 20 EC): Choose 4 out of 9.

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4205	Evolutionary Algorithms	5
Responsible Instructor	Prof.dr. P.A.N. Bosman	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Must have the ability to program your own (extensions of) evolutionary algorithms (e.g., C, C++, Java). Further, although no particular high-level prior knowledge is expected, a basic understanding of key aspects in linear algebra (e.g., matrix multiplications, inversions, decompositions, etc.), probability theory and statistics (e.g., normal distribution, statistical hypothesis testing), algorithm design (e.g., efficient local search, heuristics), complexity theory (e.g., big-O notation, algorithm analysis), will be very helpful.	
Course Contents	In this course we consider a specific subfield of Artificial Intelligence: Evolutionary Algorithms (EAs). These algorithms, sometimes also identified as being part of the class of bio-inspired algorithms, have as a metaphor the concept of natural evolution, i.e., the mechanisms by which, the fittest individuals in a population survive, reproduce, and in doing so, over time, change to be better equipped to thrive in their environment. Initiated in the 60s and 70s of the 20th century, research on EAs has progressed immensely. Today, EAs are being used to solve real-world problems in many areas, e.g. to optimize the layout of electrical wind farms, to automatically create radiation therapy treatment plans, and to optimize the architectures of deep neural networks. This course covers a spectrum of topics in EAs, ranging from basic concepts to advanced, recent, and state-of-the-art research, and ranging from theoretical to applied. In particular, topics include genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, optimal mixing evolutionary algorithms, multi-objective optimization, and real-world applications. The course is planned to have 7 lectures and 2 practical assignments. The first practical assignment pertains to experimenting with already implemented EAs on predefined problems. The second practical assignment offers more freedom, allowing you, in a group, to build your own EA (this may vary depending on student numbers and other circumstances). disclaimer: information may change depending on the developments around the coronavirus.	
Study Goals	Upon successful completion of this course, students will be able to: 1) Explain the key concepts underlying the main streams in Evolutionary Algorithm (EA) research, with in particular genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, and optimal mixing evolutionary algorithms. 2) Explain key ingredients underlying the rationale of when these algorithms work and when they do not work. In particular: schema analysis and how the match between the search bias of an EA and the fitness landscape is influenced by aspects such as variable dependencies and multi-modality. 3) Name and explain key research lines along which state-of-the-art research in EAs is done to achieve more robust, efficient, and effective EAs. 4) Identify good opportunities for using EAs, or hybrid versions thereof, in practice. 5) Properly (scientifically) experiment with EAs as well as program your own.	
Education Method	7 Lectures 2 Lab projects disclaimer: information may change depending on the developments around the coronavirus.	
Literature and Study Materials	Papers and slides that will be made available.	
Assessment	The final grade is based on 60% written exam, 40% lab practical work. disclaimer: information may change depending on the developments around the coronavirus. In particular, there may be no written exam. In that case, there will likely be several smaller practical assignments and a large practical assignment at the end of the course. Both parts will then likely count for 50%. For the large assignment there will also be chance to resit this part by means of a repair option through an oral examination of the lecture contents.	
Permitted Materials during Tests	None	
Tags	Algorithmics Artificial intelligence Optimalisation	

CS4210-A	Algorithms for Intelligent Decision Making	5
Responsible Instructor	Dr. N. Yorke-Smith	
Instructor	Dr. E. Demirovi	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Recommended: IN4010: Artificial Intelligence Techniques, or equivalent; and/or IN4301: Advanced Algorithms, or equivalent Required: basic course(s) in algorithm design and analysis, logic and probability; basic programming (in Python)	
Course Contents	Decision making is at the centre of artificial intelligence. This course gives you practical skills on a solid theoretical base. The course looks at solving mathematical models of NP-hard discrete optimisation problems. These kinds of problems lie at the heart of AI techniques such as planning, machine learning and mechanism design, and more generally combinatorial optimisation. You will learn about a range of modelling techniques from boolean satisfiability to constraint programming, and how advanced solvers for these models work. The course has plenty of real-world case studies as well as theoretical results. Apply the skills you learn in this course by taking CS4210-B: Intelligent Decision Making Project in quarter 4!	
Study Goals	By the end of this course, you will be able to identify features of real-world combinatorial decision problems, and be able to model and design systems for simplified instances of these problems using boolean satisfiability, mixed integer programming, and constraint programming over finite and real domains. You will be able to explain how SAT, CP and LCG solvers work in some detail, and how MIP solvers work at a high level.	
Education Method	Lectures, homework exercises (optional), and programming assignments. The expected workload is: 30% lectures (including preparation for the exams) 40% homework exercises (optional) 30% programming assignments	
Literature and Study Materials	Provided on Brightspace	
Assessment	The final grade depends on the grades obtained for (a) programming assignments (2 in total) [30%] and (b) the exam [70%]. The final grade is computed from the unrounded grades for the components. The final grade for the programming assignment is a uniformally-weighted average of the unrounded grades obtained for the two assignments. Programming assignments can be completed by two students working together. The exam is graded on a scale from 1 to 10. A resit will be available for the exam. The result for the exam is determined by the maximum score obtained for the original exam and the resit. In order to pass the course, the rounded grade (after resit if applicable) for each part of the course must be at least 5.0, and the rounded final grade on the course must be at least 5.8. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Algorithmics Artificial intelligence Group work Modelling Optimalisation Programming Projects Small groups	

CS4220	Machine Learning 1	5
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

CS4240	Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	Seminar Computer Vision by Deep Learning	
Expected prior knowledge	Basic pattern Recognition, Basic Machine learning, Basic statistics, Basic probability theory, Programming experience (python + numpy). Note: Without some affinity/experience with python (and numpy) programming it might still be possible to pass the course, but will become quite difficult.	
Course Contents	In this course we will look at a specific field of Artificial Intelligence and Machine Learning: Deep learning. Deep learning has shown remarkable success with large data sets and unstructured input data such as raw images/audio/text. Topics include: feed forward networks, back-propagation, optimization, convolutional nets, recurrent nets, self-attention, unsupervised methods. The course will have lectures, a seminar, a lab practical and a project: - The lectures will be on generic topics, following the book; building the backbone. - The lab assignments will have you apply basic concepts of the lecture in python notebooks. - The seminar will have students read, critique, and present relevant deep learning research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The project will have students apply and design their own (small) deep learning project in the context of scientific reproduction.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the different deep learning techniques reviewed in the course, such as SGD, MLPs, CNNs, RNNs, GANs. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [LO4]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (eg: Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) problem domain which may need to re-implement, run, evaluate, investigate, extend existing research or code [LO8] Write clearly and concisely about your code, method, results, and analysis.	
Education Method	Lectures for basic theory based on the literature Assignments: we help you to become familiar with PyTorch; applying concepts from the lecture on small problems. Lab project: design and execute your own deep learning project in the context of scientific reproductions using https://reproducedpapers.org/ Seminar: paper reading, critiquing, and presenting.	
Literature and Study Materials	Books: freely available online: - http://www.deeplearningbook.org/ - https://d2l.ai/ Research papers will be made available through Brightspace. Assignments are based on PyTorch: https://pytorch.org/	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Project: in a small group of students you work on a deep learning paper reproducibility project. 3. Exam about the papers, assignments, and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4270	Conversational Agents	5
Responsible Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Instructor	Dr. M. Bruijnes	
Instructor	Dr. P.K. Murukannaiah	
Instructor	M.L. Tielman	
Co-responsible for assignments	F. Broz	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic programming skills (e.g. Python and Java) Probability theory and statistics	
Course Contents	Chatbots, embodied and conversational virtual agents, and social robots are becoming more and more popular. Many people are owning an Alexa, Cortana or Echo or are talking to their virtual assistant on their phone. Indeed, such technologies have the potential of making our lives easier and relieve people from the more repetitive tasks. For example, it is imaginable that such systems are being used for financial applications by helping customers with frequently asked questions but also to advise them on in the long term more impactful decisions such as their pension plans. Further applications can be imagined in the area of healthcare and education, some of which are already in existence today. In this course, attention will be given to different verbal and nonverbal behavioral characteristics, like speech, intonation, gaze and gestures that humans show when communicating with both other people and machines. This behavior is then related to different dialogue functions, including turn-taking, addressing others, and backchanneling, that give shape to the communication process. This course introduces conversational agent technology. We cover agent related technologies which can be grouped into: Dialog Management NLP speech synthesis social robotics	
Study Goals	After this course you have learned to: 1) Apply relevant linguistic and psychological theory to conversational agent systems 2) Analyse human-human conversational data to better design ML models 3) Explain which components are part of a dialog system and what distinguishes rule-based from statistical dialog systems 4) Describe the design and implementation of state-of-the-art conversational agents, give examples of their application areas and analyse and discuss the limitations of current systems 5) Evaluate the effects of affect and embodiment on human-agent interaction 6) Create and evaluate a socially-aware conversational agent by applying state of the art tools and libraries	
Education Method	There are 2 lectures and 1 practicum scheduled per week. Students work in groups of 3-4 on a group project. Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours)	
Literature and Study Materials	We use the book "The conversational interface " by Michael McTear, Zoraida Callejas, David Griol. This book is freely available through the TU Delft library. https://link-springer-com.tudelft.idm.oclc.org/book/10.1007%2F978-3-319-32967-3 Other relevant material will be provided on Brightspace.	
Assessment	Online Examination (30%) Group Assignment (50%) (This assignment will result in a group report and a group online demonstration with individual question/answer part) Group presentation (20%) The exam and the assignments are both intermediate results, and will not count separately for the next academic year. Only the final grade will remain. A passing final grade for this course can only be earned when for the online examination and the group assignment at least a 5,0 is earned, and the average grade for both is at least a 5,8. Projectwork with a mark lower than 5.8 can be modified and resubmitted. The mark for resubmitted coursework will be capped to 5.8 Note that individual marks for projectwork or written exam do not carry to the next year. We further grade the labs as pass/fail. By a successful pass of all labs a bonus of 0.3 will be awarded towards the group assignment grade. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr.ir. W.P. Brinkman	

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4325	Information Retrieval	5
Responsible Instructor	Dr. C. Hauff	
Responsible Instructor	J. Yang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.	
Course Contents	Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs. Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines. Covered topics include: = Information Retrieval Models; = Indexing Techniques; = Web Search; = Information Seeking Paradigms; = Evaluation of information retrieval systems; = Components of Natural Language Processing; = Word Embedding Techniques; = Neural Language Models; = Deep Contextual Embeddings; = Neural Information Retrieval	
Study Goals	At the completion of this course, students will be able to: = Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1] = Describe and implement different indexing techniques. [Learning Objective 2] = Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3] = Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4] = Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5] = Describe typical NLP tasks and components of an NLP system. [Learning Objective 6] = Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7] = Compare word embedding and contextual embedding techniques. [Learning Objective 8] = Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]	
Education Method	Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments. Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.	
Literature and Study Materials	Scientific papers, course slides, course books - all resources are available on Brightspace.	
Books	Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA. Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press. Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson. Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing. Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".	
Assessment	Weekly individual assignment, weighting 10% of the final grade. Two group projects for IR and NLP, each weighting 45% of the final grade. The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 1 2021

AP3421	Fundamentals of Quantum Information	4
Responsible Instructor	Dr. L. di DiCarlo	
Instructor	Dr. D. Elkouss Coronas	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Knowledge of linear algebra, probability and statistics.	
Course Contents	Approximate syllabus: - quantum states, unitary operations, and measurements; - universal gate sets; - entanglement, Bell test; - basic quantum communication protocols; - basic algorithms and quantum algorithmic techniques; - basic quantum error correction; - simple physical implementations of qubits.	
Study Goals	Motivation: Quantum information is the future of computing and communication. Quantum computers offer exponential speedup over any classical computer. Similarly, quantum communication offers many advantages, including the ability to create secure encryption keys where security rests only on the laws of nature. Synopsis: This class will teach you the fundamental principles of quantum information. You will learn essential concepts that distinguish quantum from classical devices. You will learn about quantum bits and the quantum operations and measurements that can be performed on them. You will learn the basic techniques used in quantum algorithms, and examine basic examples of such algorithms. You will also take the first step in understanding how a quantum bit can be physically implemented. Aim: To learn the fundamental concepts underlying quantum computation and communication systems.	
Education Method	3 hours of lecture, 1 hour tutorial per week.	
Literature and Study Materials	The main reference textbook for the course is Nielsen and Chuang, Quantum Computation and Information, Cambridge University Press.	
Assessment	30% homework assignments, 10% in class quiz, 60% final exam. A minimum grade of 5.0 (unrounded) for the final exam is required to pass the course.	
Permitted Materials during Tests		
Continuing Courses	This course forms part of the curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information AP3421-PR Quantum Information Project CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation	

CS4070	Multivariate Data Analysis	5
Responsible Instructor	Dr.ir. F.H. van der Meulen	
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	4/4/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	

Expected prior knowledge

* Introduction Probability Theory and Statistics: see for instance

A Modern Introduction to Probability and Statistics
Understanding Why and How
Series: Springer Texts in Statistics
Dekking, F.M., Kraaikamp, C., Lopuhaä, H.P., Meester, L.E.
2005, XVI, 488 p. 120 illus., Hardcover
ISBN: 1-85233-896-2

* Basic calculus

* Linear Algebra: matrix multiplication, the inverse of a matrix, the transpose of a matrix, least square solution.
see:

David C. Lay: Linear Algebra and Its Applications
ISBN-10: 0321385179 ISBN-13: 9780321385178
©2012 Pearson)

Course Contents

PART I:
Stochastic models will be developed on the basis of probability theory. Probability theory describes the behavior of certain phenomena in terms of how likely it is that certain values will occur. Central features of the models will be discussed are random variables, probability density functions, and the expected value operator. In describing random processes and signals, the correlation function and conditional probabilities play a central role.

It addresses the following subjects:
1. Random variables. Matlab exercise on estimation of PDF, expected value and variance.
2. Refresher correlation. Calculating with correlation functions.
3. Random processes, correlation function, stationarity, wide sense stationarity, estimation of correlation function (Matlab exercise).
4. Random signal processing, power spectral density function, white noise.
5. AR processes, linear prediction: theory and Matlab exercise.
6. Markov chains.

PART II:
A course in advanced statistics about linear models, Bayesian inference, classification problems, Gaussian processes and Markov Chain Monte Carlo.

Study Goals

PART I:
1. Probability Theory
- Conditional) probabilities, the law of total probability, and Bayes rule.
- Solve probability problems that require the use of axioms of probability.

2. Definition and Description of Random Variables and Processes
PDF, PMF, CDF, Covariance, Correlation- Determine if a given PDF, PMF, CDF, variance, (auto/cross-)correlation(-function), (auto/cross-)covariance(-function), power spectral density complies with (theoretical and analytical) requirements.
- Convert the description of a probabilistic problem into a probabilistic model using PDF, PMF, or CDF.

3. PDF/PMF and Expected Value
Calculate the various forms of expected value of (combinations of) random variables and random processes
- For a given (amplitude continuous/discrete and time continuous/discrete) probability model calculate the following probabilistic (marginal, joint and conditional) characterizations: PDF, PMF, CDF, probability of an event, expected value, variance, covariance, correlation, correlation coefficient, auto/crosscorrelation function, auto/crosscovariance function, (cross) power spectral density.
- Calculate the PDF, PMF, expected value and variance of a derived random variable.

4. Properties of Random Processes
- Independence, orthogonality, uncorrelated, whiteness, IID- Determine if random variables/processes have the following properties: independent, orthogonal, uncorrelated, white, Poisson, Gaussian, Bernoulli, Markov, IID, stationary, WSS, ergodic.
- Calculate the expected value, variance, auto/crosscorrelation(function), auto/crosscovariance(function), power spectral density of a linear combination of random variables and of a linearly filtered (WSS, amplitude discrete/continuous, time discrete/continuous) random process.

5. Large NumbersCentral limit theorem, law of large numbers
- Solve problems that require the use of the central limit theorem in an engineering context
- Explain the law of the large numbers in an engineering context.

6. Statistical Estimators
- Estimated mean, variance, and correlation function
- Given a set of outcomes, sample functions or realizations, calculate estimators for expected value, variance, and (auto-)correlation function.

12. Application to Engineering Problems and Simulations
- Select and translate a simple electrical engineering or computer science problem into mathematical probability model. The emphasis is on problems in signal and image processing, telecommunication, and media and knowledge technology. The class of probability models encompasses the following random variables/processes: Bernoulli, exponential, binomial, Poisson, Gaussian, uniform.
- Justify and reflect on the approach taken in calculating or simulating (MatLab) the following probabilistic properties: PDF, PMF, expected value, variance, autocorrelation function, autocovariance function.

	PART II: After finishing this course, the student is able to apply and derive statistical methods from both the frequentist and Bayesian perspective for - linear models - classification problems - clustering problems - Gaussian process regression The student is able to give a clear presentation about the underlying statistical theory. The student is able to compute several statistical characteristics by hand.
Education Method	PART I: Lectures, working groups (problem solving), laboratory work (a Matlab exercise) Workload is around 15 hours for attending lectures, 5 hours of reading study material and preparing lectures, 15 hours for the lab course, 20 hours for preparing the exam, 3 hours for the exam, and 8 hours for a final report (66 hours in total). PART II: Classes and weekly exercises.
Books	PART I: R.D. Yates and D.J. Goodman, "Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers", ISBN 0-471-17837-3, John Wiley and Sons, New York, 2005, Second Edition. PART II: Simon Rogers and Mark Girolami "A first coruse in machine learning, 2nd edition" Chapman & Hall From this book chapters 1--4, 8 and 9 will be covered.
Assessment	The final grade is the average of the grades you get for part (I) and (II). For part (I) there is a lab and written exam, where the grade is determined by the exam, and the lab assignment should be Passed. If you fail the lab assignment, you'll get a second chance to submit around the time the resit. For part (II), there will be an on-campus written exam. To pass the course, the average should be 5.8 or higher, and the grade of each individual part should be a 5.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.
Exam Hours	PART I: Online exam of 3 hours. PART 2: On campus 3 hour written exam PART II: Written exam of 3 hours.
Permitted Materials during Tests	PART I: Self made notes on a two-sided written A4 sheet. Calculator. PART II: none
Remarks	PART II: This course is particularly interesting for students that are interested in statistical exploratory and quantitative techniques to analyse multivariate data.

CS4200-A	Compiler Construction	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	6/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	- programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-A covers the following topics: * Syntax and parsing - concrete syntax, abstract syntax - context-free grammars - derivations, ambiguity, disambiguation, associativity, priority - parsing, parse trees, abstract syntax trees, terms - pretty-printing - parser generation - syntactic editor services * Static semantics and type checking - name binding, name resolution, scope graphs - types, type checking, type inference, subtyping - unification, constraints - semantic editor services	
Study Goals	After this course, students should be able to: * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define the syntax of a programming language using declarative syntax definition that describes the concrete and abstract syntax of a programming language * Define basic editor services * Define the type system (name binding and typing rules) of a programming language using constraint generation rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the front-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%) - course project (50%)	

	Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	No materials are permitted during the exam.
Judgement	to be decided

CS4215	Quantitative Performance Evaluation for Computing Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	Todays computing systems become ever complex, due to the rapid development of hardware and software technology. It is challenging to design and run computing systems that guarantee users performance requirements in a resource efficient way. Various quantitative methods are applied to capture such complex system dynamics and predict metrics of interests, from the designing phase of the systems to the runtime performance, e.g., job response times and system anomaly. To optimize the performance of computing systems, a deep understanding on those methods and their applications on the system design are essential. Having practical hand-on experience on designing experiments, deriving models, and validating results with benchmark systems will prepare students to tackle challenges of real world computing systems.	
Study Goals	LO1. Design full/fractional factorial experiments for multi-variate regression analysis, e.g., finding critical parameters for deep learning clusters LO2. Apply queueing theory to analyse and predict run-time performance of applications, e.g., the average response times of on-line ML training service LO3. Apply machine learning models to analyse and predict the system dependability, e.g,root cause analysis for machine failure. LO4. Conduct experiments to profile applications and extract their workload parameters on real systems, e.g., e.g., deep learning clusters LO5. Develop resource management policies and validate them on real computing systems, e.g., deep learning clusters	
Education Method	Lectures: 7 weeks X 2-4h Practical: Derive, validate and evaluate performance models and resource management strategies for a chosen system via homework and group project. Multiple types of computing and network systems can be chosen from. Deliverables include a report and group presentation.	
Books	1) Performance Modeling and Design of Computer Systems: Queuing Theory in Action by Mor. Harchol-Balter 2) Design and Analysis of Experiments by Douglas C. Montgomery, 3) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics.	
Assessment	Homework (40%): 2 individual homework Group project (60%): group project report and presentation disclaimer: information may change depending on the developments around the coronavirus.	

CS4270	Conversational Agents	5
Responsible Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Instructor	Dr. M. Bruijnes	
Instructor	Dr. P.K. Murukannaiah	
Instructor	M.L. Tielman	
Co-responsible for assignments	F. Broz	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic programming skills (e.g. Python and Java) Probability theory and statistics	
Course Contents	Chatbots, embodied and conversational virtual agents, and social robots are becoming more and more popular. Many people are owning an Alexa, Cortana or Echo or are talking to their virtual assistant on their phone. Indeed, such technologies have the potential of making our lives easier and relieve people from the more repetitive tasks. For example, it is imaginable that such systems are being used for financial applications by helping customers with frequently asked questions but also to advise them on in the long term more impactful decisions such as their pension plans. Further applications can be imagined in the area of healthcare and education, some of which are already in existence today. In this course, attention will be given to different verbal and nonverbal behavioral characteristics, like speech, intonation, gaze and gestures that humans show when communicating with both other people and machines. This behavior is then related to different dialogue functions, including turn-taking, addressing others, and backchanneling, that give shape to the communication process. This course introduces conversational agent technology. We cover agent related technologies which can be grouped into: Dialog Management NLP speech synthesis social robotics	
Study Goals	After this course you have learned to: 1) Apply relevant linguistic and psychological theory to conversational agent systems 2) Analyse human-human conversational data to better design ML models 3) Explain which components are part of a dialog system and what distinguishes rule-based from statistical dialog systems 4) Describe the design and implementation of state-of-the-art conversational agents, give examples of their application areas and analyse and discuss the limitations of current systems 5) Evaluate the effects of affect and embodiment on human-agent interaction 6) Create and evaluate a socially-aware conversational agent by applying state of the art tools and libraries	
Education Method	There are 2 lectures and 1 practicum scheduled per week. Students work in groups of 3-4 on a group project. Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours)	
Literature and Study Materials	We use the book "The conversational interface " by Michael McTear, Zoraida Callejas, David Griol. This book is freely available through the TU Delft library. https://link-springer-com.tudelft.idm.oclc.org/book/10.1007%2F978-3-319-32967-3 Other relevant material will be provided on Brightspace.	
Assessment	Online Examination (30%) Group Assignment (50%) (This assignment will result in a group report and a group online demonstration with individual question/answer part) Group presentation (20%) The exam and the assignments are both intermediate results, and will not count separately for the next academic year. Only the final grade will remain. A passing final grade for this course can only be earned when for the online examination and the group assignment at least a 5,0 is earned, and the average grade for both is at least a 5,8. Projectwork with a mark lower than 5.8 can be modified and resubmitted. The mark for resubmitted coursework will be capped to 5.8 Note that individual marks for projectwork or written exam do not carry to the next year. We further grade the labs as pass/fail. By a successful pass of all labs a bonus of 0.3 will be awarded towards the group assignment grade. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr.ir. W.P. Brinkman	

EE4C06	Networking	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	PART 1: Basics, concepts and computations of networks 1. Basics of networking & introduction to Network Science - what is a network? Representation of a graph, basics of graph theory, overview of the relatively new theory of complex networks, called Network Science. - important characterizers of a network (network metrics) - basic network/graph models - examples of real-world networks (airline transportation, the web and Internet, social networks, brain networks, etc.) and applications of network science 2. Concepts of networking - routing - Quality of Service (QoS) - traffic management and scheduling - network robustness (failure, cascading effects,...) - overlay networking and new aspects of networking such as interdependent networks PART 2: Applications and examples of networks (as listed below) will be taught (some of those by a guest lecturer). Ranging from year to year, a selection among the following will be covered: 1. Electrical networks (smart grids) 2. Networks on Chip (NoC) 3. Optical networks 4. Computer Networks (the Internet) 5. Mobile communication networks 6. Sensor networks 7. Biological networks 8. Social networks	
Study Goals	The course on Networking aims to provide a general and basic introduction to the art of networking, that tries to unravel the operation and behavior of networks, both man-made (infrastructures such as the Internet and power grids) as well as networks appearing in nature (such as the human brain, biological networks and social human interactions). The course on Networking will introduce concepts of the new Network Science, that basically studies the interplay between, on the one hand, the processes (also called functions or services) on the network and on the other hand, the underlying topology, that is mostly changing over time as an evolving organism, rather than as given or fixed object. Network Science combines many disciplines such as graph and network theory, probability theory, physical processes, control theory and algorithms. After this course, students are expected to represent/abstract real-world infrastructural network (e.g. a communication system) as a complex network, understand the basic methods to analyze properties of networks and dynamic processes on networks. Students will also understand why processes on networks and design of networks are so complex. Finally, students may appreciate the fascinatingly rich structure and behavior of networks and may realize that much in the theory of networks still lies open to be discovered.	
Education Method	Lectures, slides & homework	
Assessment	written examination disclaimer: information may change depending on the developments around the coronavirus.	

ET4388	Ad-hoc Networks	5
Responsible Instructor	Dr. R.R. Venkatesha Prasad	
Contact Hours / Week x/x/x/x	3/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Wireless communications and networking Computer communication principles, Layering principle of Computer Networks. Digital communication.	

Course Contents

IMPORTANT NOTICE

Please note that the prevailing conditions of the COVID19 may force us to modify the study guide information. It may change depending on the developments around the coronavirus. We may have to make changes to the teaching methodology, assessment, practical work, assignment and group activities. This will be instructed by the faculty/university management based on the orders of the government from time to time to protect the faculty and students.
The above applies to all the fields in this coursebase for this course.

The course will be offered ONLINE. Face to Face meeting may be possible depending on the situation [Safety of everyone is the highest priority].

Ad-hoc networks are formed in situations where mobile computing devices require networking applications when a fixed network infrastructure is not available or not preferred to be used. In such cases, mobile devices may possibly set up an ad hoc network themselves. Ad-hoc networks are decentralized, self-organizing networks and are capable of forming a communication network without relying on any fixed infrastructure.

Ad-hoc networks form a relatively new field of research. In this lecture, besides general introduction to ad-hoc networks and their applications, we will focus on state-of-the-art methods and technologies for forming an ad-hoc network and maintaining its stability despite the dynamics of the network.

The contents of the course are as follows:

Positioning and applications (Chapter 1, 2 & 3 of the textbook, these topics are basics & pre-requisites; And Chapter 5)
o Definition of ad-hoc networks
o Comparison with infrastructure based systems
o Typical applications
o Advantages and challenges
o Radio technologies for ad-hoc networks
o Wi-Fi, Zigbee, Bluetooth

Modelling ad-hoc networks
o Propagation models
o Topology models based on graph theory
o Degree and hopcount
o Connectivity theorems

MAC protocols for ad-hoc networks (Chapter 6, 10 of the textbook)
o Introduction to MAC protocols
o Issues and design goals
o Classification
o Directional, muli-channel MAC protocols
o Energy efficiency in MAC protocols
o Quality of service

Self organisation and Routing (Chapter 7, 8, 11 of the textbook)
o Flooding
o Node discovery, neighbour discovery
o Route establishment
o Topology maintenance, localisation
o Proactive, reactive and hybrid routing
o Typical protocols
o Energy efficiency in routing
o Broadcast and multicast
o Effects of mobility on connectivity and capacity
o Effect of nodes joining and leaving the network

Advanced issues in ad hoc networks
o Wireless sensor networks (Chapter 12 of the textbook and papers)
o Cooperation (Reference papers)
o Simulating ad hoc networks as part of project (optional: ns3, OMNET, OPNET)
o Energy Harvesting

Project presentations by students

Study Goals

By the end of this course students should be able to:
- Model the ad-hoc networks using Graphs.
- Describe the working principles of medium access control protocols for ad-hoc networks
- Explain the working principles, advantages and disadvantages of different classes of routing protocols for ad-hoc networks
- Choose various components to form a coherent ad hoc networking architecture
- Develop a simulator to evaluate the MAC and routing protocols for ad hoc networks
- Assess the suitability of ad-hoc networks for different communication needs and scenarios

Education Method

The course will be taught in lecture form. The presence of students at all lectures is required for optimum result. Students are required to participate actively in various forms of activities and peer-learning. New forms of teaching aids are used.

Literature and Study Materials

1. Textbook: Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004.

	2. Lecture notes consisting of slides presented at the lectures (Slides are only teaching aid and they are not substitute for textbooks, research papers, etc). 3. Some recent journal papers 4. Optional Reference Books 4.1. Distributed Algorithms, Nancy A. Lynch, Morgan Kaufmann, 1996 (for networking algorithms) 4.2. Ad Hoc Mobile Wireless Networks, Principles, Protocols and Applications by Subir Kumar Sarkar , C Puttamadappa , and T. G Basavaraju, Auerbach Publications, 2008. This book is avaliable online in the library. 4.3. Wireless Ad Hoc and Sensor Networks, A Cross-Layer Design Perspective by Jurdak, Raja, Springer, 2007. This book is avaliable online in the library. 4.4. Ad-hoc Networks: Fundamental Properties and Network Topologies, by Ramin Hekmat, Springer. 5. OPNET/ns-2 web pages, tutorials and video lectures
Books	Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004. However, I also use other materials from Internet and other books listed above.
Assessment	1. There will be written tests/examinations for this course. 2. The students will carry out a project in a group and submit a short report. 3. Participation in off-track discussions on Facebook/Brightspace/FeedbackFruits and wikis. Final score is based on marks obtained during tests, project, assignment (in groups) and bonus marks. All the details will be given in the first class. Breakup: 2 Tests + Final Exam = 55% Project 40% Self assessment + Reflection 3% Activities on Feedback Fruits or any Online platform 2% ====================== Changes due to COVID19 in 2021 ================ There are three chapters Chapter 1: Network modelling Chapter 2: MAC protocol Chapter 3: Routing + Lab: Simple experiments using laptops (individual) + simulations (group-wise) Marks Breakup -------------- Homework/Assignments/Group works Part A1: Group-wise assignment. 3 times (1 per chapter) -- 15 marks Part A2: Group-wise Q&A. 3 times (1 per chapter) -- 30 marks Part B1: Individual experiment+report Part B2: Group-wise simulations + report + demo Part B1 + Part B2 - 50 Marks Part C: Self-assessment + peer activities -- 5 marks Resit: Part A1 & A2 will not be repeated. Only Part B1 & B2 are allowed for the Resit this year, because of COVID19; and the projects should be done individually in Resit. == (More information will be given in the first class) disclaimer: Information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	Different conditions for different test/exams. Conditions will be informed 1 week before the exams/test.

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

IN4049TU	Introduction to High Performance Computing	6
Responsible Instructor	Prof.dr.ir. H.X. Lin	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear algebra (matrix and vector operations), numerical analysis (solution of a system of linear equations;some experience with a programming language (e.g., C) is preferred but not required.	
Course Contents	This course is intended for students who are interested in computing-intensive research. In the course, a number of algorithms that are being used within a diversity of research areas is considered. The scaling behaviour of these algorithms in case of an increasing problem size and/or an increasing number of processors, is analysed. Attention is paid to those aspects of computer architectures that are important to understand the resulting performance, such as the memory hierarchy and the interconnection network. By analysing a number of case studies (applications) with respect to their computing-intensive character, possible bottlenecks will be determined. Based on performance analysis, it will be indicated how the effect of those bottlenecks can be reduced. The goal is to learn how to get a high performance with the available hard/architecture. The lab exercises will be done on a cluster of computers, the DAS-5 system at TU Delft. The emphasis will be on designing efficient parallel algorithms and on the necessary optimalisation of the performance. During the lab exercises, the following types of problems will be elaborated on: a parallel Poisson solver, a parallel finite element simulation and a parallel N-body simulation. More information, such as handouts and slides, can be found the Brightspace.	
Course Contents Continuation	High Performance Computing, parallel programming, parallel algorithm	
Study Goals	1. Knowledge about high performance computer systems including parallel and distributed architectures, and programming models; 2. Basic knowledge about the concepts of data decomposition and parallel algorithms; 3. Knowledge about various high performance (numerical) methods and their parallelization; 4. Capable to implement parallel programs (using MPI) on cluster of computers and GPU (using Cuda); 5. Obtain some experience on performance analysis of parallel programs.	
Education Method	Lectures, computer lab exercise using MPI. As an option, answers to the bi-weekly quizzes can be handed in, and a maximum of one bonus point to the exam grade can be obtained.	
Computer Use	Lab exercises (mandatory): implementing (small) parallel programs with C, MPI and Cuda.	
Literature and Study Materials	Will be made available throughout the course and can be downloaded from the Brightspace.	
Assessment	Written exam (50%) + Lab work (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Via Osiris	

IN4191	Security and Cryptography	5
Responsible Instructor	Z. Erkin	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	UT-201500042 Privacy Enhancing Technologies (Q4)	
Expected prior knowledge	Basic understanding on the following is suggested. -Probability and statistics -Programming skills	
Course Contents	Motivation: Computers are now found in every layer of society, and information is being communicated and processed automatically on a large scale. Examples include medical and financial files, automatic banking, video-phones, pay-tv, teleshopping and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy. Synopsis: Security and cryptography are essential components of any digital system. In this course, the fundamentals of secure data storage and transportation of information are described. In particular, classical (e.g. Caesar, Vigenere) and modern encryption schemes (RSA, DES, AES, Elliptic curves) are described along with their mathematical background such as number theory. Methods for authentication, data integrity and digital signatures are discussed in detail, as these are the main components of many security architectures. The course also investigates more advanced topics such as zero-knowledge proofs and secret sharing schemes. Aim: It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security and privacy, as well as is familiar with present applications. Learning outcomes: The goal is to make students familiar with the basic concepts applied cryptography, including classical cryptography and modern secret key and public key cryptography. In particular, the students will acquire A sound understanding of the notion of security An understanding of the confidentiality, integrity and authenticity needs of the society Understand the role of cryptographic primitives including the differences between symmetric and asymmetric cryptography, the role of hash functions, digital signatures and PKI Understand the advanced topics in cryptography needed for the modern society with untrustworthy entities Among others things, the following topics are covered: -Classical systems -Information theoretic security -Definition of Security notions -Symmetric encryption (e.g. DES, AES) -Asymmetric encryption (RSA, Elliptic Curves) -Hash functions -Random number generation -Key Management -Digital Signatures, -*Secret Sharing. (if time permits) -*Zero Knowledge proofs (if time permits)	
Study Goals	It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security, as well as is familiar with present applications. Through assignments, students are expected to have the chance to work on the topics covered in the lectures.	
Education Method	Lectures, assignments and weekly exercises. Attention: This course requires full effort of 140 hours. Even more, if you lack the background (probability and modular arithmetic) Planned Workload: Lectures: 28 x 45minutes sessions, total 22 hours Practice session: 7 x 90 min. total 12 hours Assignments: 3 x 20 hour, total 60 hours Weekly study: 7 x 4 hours, total 28 hours Exam preparation: 20 hours Exam: 3 hours	
Literature and Study Materials	Cryptography made simple, Nigel P. Smart, 2nd Edition, Springer, 2016 (PDF Available Online) Handouts of lectures	
Assessment	Written exam(70%) + mandatory assignments (30%). Passing grade for the written exam and assignments is required. Please refer to CS regulations for further details. There is NO reparation for assignments. Points from previous years cannot be transferred. NOTE: the study guide information may change depending on the developments around the COVID-19 pandemic	
Exam Hours	In case of in person examination at campus: The exam is closed book. A cheat sheet of size A4, hand written is allowed for the written exam. Name and student number has to be present on each side. In case of remote exam from home: open book: textbook, slides and self-made notes only. randomised/customised exam.	

Permitted Materials during Tests	Only non-scientific calculators.

IN4252	**Web Science & Engineering**	**5**
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4307	Medical Visualization	5
Responsible Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0 lectures & 0/4/0/0 lab.	
Education Period	1 2	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic knowledge of linear algebra, calculus and programming is needed. This course (IN4307) has been designed to complement the courses Advanced Image Processing (ET4283) and Medical Imaging (AP2231TUD). However, these two courses are NOT pre-requisites.	
Course Contents	Theory and practice (Notice project extends to Q2) of medical visualization. This includes the following aspects: data acquisition basics, clinical practice; image processing, e.g., filtering, segmentation and measuremen; medical volume visualization; illustrative visualization; advanced visualization for complex modalities; interaction techniques for medical data; advanced applications.	
Study Goals	By the end of the course, you should be able to LO1: Explain medical visualization algorithms and their applicability to medical problems. LO2: Discuss the advantages and disadvantage of medical visualization algorithms. LO3: Build a medical visualization system for a given problem: a. Discuss a suitable visualization for a given medical problem. b. Implement the most suitable solution. c. Judge the performance of the implemented solution.	
Education Method	The course will be based on a combination of lectures and practical assignments. A final project will be developed in Q2	
Literature and Study Materials	Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications Bernhard Preim and Charl P. Botha (not mandatory)	
Assessment	The evaluation will be based on - a written (or oral if the number of students allows) exam (40%) - assignments during the semester (10%) - a final project (50%) The final project will be done during the 2nd quarter. Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The assignments will consist of small programming exercises and open questions, as preparation for the final project. The practical sessions will provide time for working on the assignments with assistance. The deliverables will be program code and/or answers to questions. The final project will be the design and implementation of a visualization system for a given medical problem. The final project will be carried out in teams. The deliverables for the final project will be a report (paper), the results (e.g., code) and a short video presenting the project (i.e. screencast). The written exam will be arranged at the end of the first quarter. You are allowed to have the slides and material of the course during the exam. No computer or laptop is allowed. The exam has a resit. The project will have a resit if the exam (NOT the resit exam) has been passed with a mark of 7.5 or higher and it will be on an individual basis. The project resit is not automatic and must be initiated by you within two weeks of the grades being published. Resit of a project will mean starting a new project. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	Notes and written material. No computers.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

IN4387	System Validation	5
Responsible Instructor	C.B. Poulsen	
Contact Hours / Week x/x/x/x	4,0,0,0 Lectures & 2,0,0,0 lab	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	Embedded Systems Masters	
Expected prior knowledge	There are no strict entry conditions for this course. However, prior knowledge of requirements analysis is recommended. Furthermore, a good basic knowledge about logic and set theory is extremely beneficial.	
Parts	Behavioural specification of sequential and parallel using labelled transition systems, process algebra, and abstract data types; model checking of such systems using the modal mu-calculus. Model-based testing.	
Summary	Everyone who ever designed an embedded system or a communication protocol involving several components executing simultaneously has experienced that such software is inherently susceptible to bugs. Typical problems that occur are race conditions, deadlocks, and unexpected interplay between different components. Due to the parallel nature of these systems, it is notoriously hard to detect such bugs using testing (for example, timing plays a crucial role). The following quote from the famous Dutch computer scientist Edsger W. Dijkstra illustrates a further problem with testing. Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence. Edsger W. Dijkstra In this course, we study model checking, which in contrast to testing can also be used to show the absence of bugs. Model checking is a technique in which we consider all states in (a model of) the system based on an abstract model. Based on this state space we verify whether the model satisfies the desired properties. Properties are typically derived from the requirements of the system. We will restrict ourselves to verification techniques that do not reason about timing (merely about the order in which event happen). Finally, we see how model-based testing can be used to show that an implementation conforms to the specification of the system.	
Course Contents	Behavioral Specification using Process Theory (Labelled Transition Systems, various notions of behavioral equivalence) and process algebra. Model checking the modal mu-calculus, and model-based testing using IOCO.	
Study Goals	Upon completion of the course: 1. The student knows the fundamental theory necessary for specifying the behavior of embedded systems and for reasoning about this behavior. 2. The student can describe simple systems using this theory. 3. The student can formally specify requirements and prove (or disprove) them on the behavior. 4. The student is able to model a concrete embedded system, and verify that it satisfies its requirements. 5. The student is able to show that an implementation of a system conforms to its specification.	
Education Method	Lectures + Programming Assignments + Practical Project The course is structured into two parts: 1. There will be weekly mandatory programming assignments in the first four weeks of course will be a small set of mandatory. The programming assignments are assessed as pass/fail. The programming assignments are due after the first four weeks of the course. 2. In the last four weeks of the course, you will self-organize into groups of (about) 4 students, and will develop and verify a model of an embedded system. You will write a report that documents your model and its development. There will be a written exam with programming assignments at the end of the course.	
Computer Use	The theory introduced in this lecture is at the heart of the mCRL2 tool set. This tool set can be used to specify and verify systems, and visualize them. To be able to carry out the project it is required that the mCRL2 tool set is installed on your laptop (or one of the TU Delft systems, if you do not have a laptop you can use). It is open source software, and is free of charge. The software can be obtained from https://www.mcrl2.org.	
Literature and Study Materials	The course is based on the book by Groote and Mousavi (see "Books"). All other materials will be published on Brightspace.	
Books	J.F. Groote and M.R. Mousavi. Modeling and Analysis of Communicating Systems. MIT Press, 2014. ISBN: 9780262027717 (Chapters 1-7,11 are mandatory)	
Assessment	The result of this course will be based upon the results of the written examination (50%) and the practical project (50%). For both the programming exam and the practical project, a minimum of 5.0 is required in order to pass the course. To be eligible for taking the exam you must submit and pass the mandatory programming assignments for the first four weeks of the course. Grades of the project or written exam do not automatically carry over from previous years, so upon retaking the course talk to your lecturer first. For the exam a resit is scheduled. Please note that the study guide information for this course may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	The exam will be a 3 hour written exam with programming questions. You are allowed to use the book and any other static resources. You are not allowed to communicate or discuss exam questions with anyone but members of the teaching team for the course. Discussing or copying code will be considered fraud, and is reason for expulsion from the course.	
Enrolment / Application	Brightspace	
Co-Instructor	E. Visser	

WM-ITAV-4010	Scientific Writing	2
Module Manager	L. Meester	
Instructor	A. Glasbergen-Plas	
Instructor	M. Looij	
Instructor	M.J.Y. Wackers	
Instructor	S. Baars	
Instructor	M. Bliekendaal	
Instructor	L.C. Schroten	
Instructor	Drs. W.J. Blokzijl	
Instructor	Drs. B.M.D. van der Laaken	
Instructor	Drs. P.C. Post	
Instructor	Drs. A.E. Kam	
Co-responsible for assignments	Drs. B.M.D. van der Laaken	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1 2	
Exam Period	none	
Course Language	English	
Course Contents	In this course, you learn to write a scientific article, either a research article based on your own research data or a literature review about a subject of your own choice. This is a necessary skill for anyone who wants to pursue an academic career after their graduation, but it can also be used immediately for all academic texts you will write during your Master programme, such as your Master thesis. In seven weeks, we will go through all steps of the writing process, from formulating a good main question and finding relevant literature, to the actual writing, re-writing and final editing. You will exercise with finding, reading and managing relevant academic literature, writing in an academic style, building a comprehensive argumentation, reviewing fellow students' articles and using other students and the instructor's comments to improve your own work.	
Study Goals	The purpose of this course is to learn how to write a scientific text. To achieve this, at the end of this course you will: know what the main characteristics are of a scientific text be able to formulate a main question be able to find, critically read and manage scientific literature be able to use literature properly and avoid plagiarism be able to build up your argumentation be able to structure an article according to the conventions in your field of study be able to use scientific English style be able to use tables and figures to support and communicate your results be able to give feedback on somebody elses article be able to use feedback for improving your work.	
Education Method	(Online and/or on campus) practical, in 6 sessions (attendance mandatory). Every week you have to read some background information about scientific writing and hand in a part of your text. Participants must attend all sessions - one missed session is allowed only - and hand in all assignments in time. Students who receive a pass for this course are rewarded with 2 ects. This equals 56 hours of study. A total of 12 hours is spent on attending (online) classes, in which you can ask questions, discuss the feedback you received on your work and discuss aspects of scientific writing with your fellow students; the remaining 44 hours is for self study, writing and revising. In seven weeks, from preparing lecture 2 up to handing in your final article in week 8, you will have to spend at least 6 hours of self study on this course, every week. It is important that you make sure you have this time in your personal schedule. At the beginning of the course you will mainly be reading up about scientific writing and the subject of your text. As the course proceeds, you will be spending more of your time on writing, giving feedback and revising your own text.	
Books	Theory about academic writing will be made available through Brightspace.	
Assessment	You write a scientific article of 3000 words (excluding the list of references and the abstract, 10% margin in the word count). You have to hand in a parts of your article every week. Your final grade is based on the final article. An evaluation form for the grading of the article is available throughout the course.	
Elective	Yes	
Category	MSc level	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 2 2021

CS4015	Behaviour Change Support Systems	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	none	
Course Contents	Behavior change support systems (BCSS) are computer-based systems that support individuals to form, alter or reinforce cognitions, attitudes or behaviors without using coercion or deception. They can serve individuals throughout the various stages of a change process, such as awareness developing, contemplation, action strategy development, development of new behaviors, and maintaining these new behaviors. Virtual healthcare coaches, negotiation support systems, and applications that provide individuals with personalize financial guidance are three examples of these systems. To establish, modify or maintain change BCSS can deploy computerized persuasive strategies (e.g. reducing effort to establish target behavior, or argumentation and reflection strategies), simulations (e.g. serious gaming, virtual reality), relational software agents (e.g. ePartners, virtual coaches), and personalization based on longitudinal user data. BCSS are found in many domains, including education, sales, negotiation, management, and particular in the health domain.	
Study Goals	The course allows students to achieve understanding of principles, concepts and theories underlying BCCS systems and methods for designing them.	
Education Method	In the pre-recorded video material, theories, principles and methods are presented, discussed and illustrated with examples from the field. The video material is support by online self-tests. In the lectures, teacher and students discuss and make assignment related to pre-recorded material of that week. At home students work on their own in small groups on coursework assignments to develop a product design for a BCSS. In the practicum session student groups presented the progress on their coursework and receive feedback. Expected Workload Pre-recorded video material: 18 hours (2 hours × 9 lectures) lecture: 10 hours (10 × 1 hours) practicum 14 hours (7 × 2 hours) Reading time: 36 hours (9 × 4 hours reading time) Preparation presentation: 3 hours (3 × 1 hours for each presentation) Coursework project, including writing report, and final presentation: 50 hours Exam preparation and revision: 9 hours Total = 140 hours	
Literature and Study Materials	Will be announced on brightspace	
Books	Wendel, S. (2013). Designing for Behavior Change: Applying Psychology and Behavioral Economics. " O'Reilly Media, Inc.".	
Assessment	The course is assessed by coursework and an exam as follows: (60%) computerised examination (or oral exam) (40%) Coursework Project (resulting in a report, and final presentation include question and answer round where individual group members are assessed on coursework) If the expected number of students registering for exam or resit is small, the teacher might decide to replace the computerized examination by an oral examination. Separate marks will be given for exam and coursework, only a combined mark is recorded in Osiris. A passing final grade for the course can only be earned when for both the exam and coursework at least a 5.0 is earned, and the weighted grade for exam and coursework is at least a 5.8. Resit coursework A second submission of modified coursework is only allowed for coursework that received a fail mark (<5.8) for the first submission. Overall resit mark for coursework will be capped to 5.8. Note that individual marks for coursework or computerised exam (or oral exam) do not carry to the next year. disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	2	
Co-Instructor	M.L. Tielman	

CS4055	High Performance Data Networking	5
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Basic understanding of networking and programming (ideally Python).	
Course Contents	The Internet has become of critical importance to society. However, the large size of networks and abundance of protocols have made network management very complex. The novel concept of network programmability addresses this complexity and has resulted in a paradigm shift in how networks are (or can be) operated. The high-performance data networking course is an advanced networking course that will introduce you to the concept of network programmability and which treats fundamental networking concepts like Quality of Service and network resilience.	
Study Goals	The learning objectives of this course are twofold: (1) The student should gain knowledge of the treated networking technologies. (2) The student should be able to apply and work with the programmable network technologies in a network emulator (Mininet).	
Education Method	Approximately 50% of the course will consist of lectures and selfstudy and 50% focuses on (homework) exercises and instruction classes.	
Literature and Study Materials	Slides and a reader containing the exercise material.	
Assessment	The final assessment will be based on an exam that covers both the theory from the slides as well as the content from the reader.	
Remarks	Disclaimer: The information about CS4055 (including its assessment) may change depending on the developments around the coronavirus.	

CS4090	Quantum Communication and Cryptography	5
Responsible Instructor	Dr. S.D.C. Wehner	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear Algebra, Probability & Statistics, Q101 (Fundamentals of quantum information)	
Course Contents	This class will introduce you to the fascinating field of quantum communication! We will look at the state of the art of quantum networks, and explore techniques for building quantum repeaters that promise to deliver qubits over long distances. We also briefly look at one of the most famous application of quantum cryptography, quantum key distribution. Caution: 1. This class requires you to take "Fundamentals of Quantum Information" in Quarter 1 2. The focus of this class is presently on quantum communication, and we will only briefly look at quantum cryptography. As such, this class is not held in flipped classroom style in conjunction with edX QuCryptoX as in previous years.	
Study Goals	The student will acquire: A good understanding of the fundamental concepts of quantum communication Insight into the differences between classical and quantum communication and cryptography Skill set required to follow the remainder of the quantum curriculum (Q301 Quantum hardware and Q401 Quantum electronics)	
Education Method	Lectures and tutorials. If remote classes continue this fall, then recorded lectures and live discussion session	
Literature and Study Materials	Primary: Slides Review Articles Auxilliary: Nielsen and Chuang Quantum computation and information, Cambridge University Press.	
Assessment	Homework (70%) and Final Project (30%) disclaimer: information may change depending on the developments around the coronavirus.	
Tags	Abstract Adventurous Algorithmics Challenging Group Dynamics/Project Organisation Information & Communication Integrated Intensive Involved Lineair Algebra Mathematics Physics Quantum Signals Technology Telecommunication	

CS4200-B	Compiler Construction B	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/6/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	- CS4200-A: Compiler Construction A (recommended) - programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Note that the title of this course should be "Compiler Construction B", not "Compiler Construction Project". The course combines theory and practice of compiler back-ends. Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-B covers the following topics: * Transformation - rewrite rules, rewrite strategies - simplification, desugaring * Dynamic semantics and code generation - operational semantics, program execution - virtual machines, assembly code, byte code - code generation - memory management, garbage collection * Data-flow analysis - control-flow, data-flow - monotone frameworks, worklist algorithm	
Study Goals	After this course students students should be able to * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define transformations on abstract syntax terms to simplify programs * Define a code generator that translates source language abstract syntax trees to object language instructions using rewrite rules * Define data-flow analyses using control-flow and data-flow rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the back-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%)	

	- course project (50%) Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	not applicable
Judgement	to be decided

CS4220	**Machine Learning 1**	**5**
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

IN4089	Data Visualization	5

Responsible Instructor: T. Höllt

Instructor: Prof.dr. E. Eisemann

Contact Hours / Week x/x/x/x: 0/2/0/0 & lab

Education Period: 2

Start Education: 2

Exam Period: 2
3

Course Language: English

Required for: Master course MKE/ST/DS

Expected prior knowledge

IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required).
The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.

Course Contents

Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization,
The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines.

As a computer science course, affinity to algorithmic thinking and programing skills will be needed.

Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.

Study Goals

The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data.
Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more.

By the end of the course, you should be able to
LO1: Discuss a large range of visualization techniques.
LO2: Discuss a perception principle of visualization.
LO3: Explain mathematical principles and algorithms of visualization techniques.
LO4: Design suitable visualization systems for a given practical data analysis problem.
LO5: Implement visualization systems for a given practical data analysis problem.

Education Method: Lectures, practical assignments, self-study, and projects.

Literature and Study Materials

Course slides, instructions for projects, and selected literature.

Chapters from:

Visualization Analysis and Design
Author: Tamara Munzner
CRC Press

Visual Computing for Medicine
2nd Edition
Theory, Algorithms, and Applications
Authors: Bernhard Preim Charl Botha
Morgan Kaufmann

All available in electronic form via Brightspace or at TUDelft library.

Assessment

The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation.

disclaimer: information may change depending on the developments around the coronavirus.

Special Information

It is necessary that you register/enroll on Brightspace for this course.

In the first lecture, details on the evaluation and practical information on the course will be given.

Judgement

The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam.

The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project.

Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions.

The project is evaluated based on the developed result, its documentation and presentation.

Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam

Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6))

The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.

IN4150	Distributed Algorithms	6
Responsible Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Knowledge of Computer Networks (CSE1405) and Operating Systems (CSE2430) is useful as background for understanding this course.	
Course Contents	Introduction to distributed algorithms; notions of time and ordering of events; distributed algorithms for message ordering, detecting global states, termination detection, deadlock detection, mutual exclusion, election, minimum-weight spanning trees, fault tolerance, consensus, and agreement; blockchain technology and its relation with consensus.	
Study Goals	Understand the main fundamental problems in distributed systems Understand the most important distributed algorithms that solve these problems Be able to reason about the execution of distributed algorithms Be able to program distributed algorithms Be able to select and summarize relevant literature on distributed algorithms	
Education Method	Lectures and lab work executed in groups of two students	
Literature and Study Materials	Lecture notes and lecture slides (available on Brightspace)	
Assessment	One paper summary and an in-person, on-campus written exam (closed book), which together are an integrated whole. The grade of this exam, which includes the grade of the paper summary (with a weight of 25%), is the grade for the course. In case in-person, on-campus written exams are not possible because of COVID-19, the written exam will be replaced by individual online oral exams.	
Permitted Materials during Tests	None except the list of algorithms	
Remarks	Lab work is 40 hrs.	

IN4302TU	Building Serious Games	5
Responsible Instructor	Dr.ir. A.R. Bidarra	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2	
Course Language	English	
Expected prior knowledge	For CS students: programming experience with some object-oriented language; experience with graphics, AI and/or some game engine(s) is a plus. For all students: though not compulsory, it may be convenient to have followed the course SEN9235 (Game Design Project), which is taught in the first quarter.	
Course Contents	Project-based interdisciplinary course, open to MSc students of all faculties. The main goal of the project is to take students with varying talents, backgrounds, and perspectives and put them together to do what none of them could do alone: to design and implement a serious game aimed at being applied in a real-world setting (educational, social, training, health-related, etc.). The emphasis is both on constructively fulfilling the game requirements, and on deploying the adequate technology for that purpose. Assignments for this course will be provided by real-world end-users (e.g. companies or the Science Centre Delft), to whom the group will be reporting throughout the term of the project.	
Study Goals	At the end of the project, the student will demonstrate proficiency in the following aspects: o identifying and valuing the soft skills necessary to work in interdisciplinary teams o responsibly interacting within a team, integrating its members' varying talents and expertise o adapting with flexibility to the dynamic requirements of a complex external assignment o translating feedback received into proactive personal development steps Additionally, the CS student will demonstrate proficiency in the following specific aspects: o identifying, selecting and deploying the most adequate game technologies for the given serious game domain and constraints o deepening programming skills while building a complex and large software system in an agile context	
Education Method	Project: teams work intensively as a small game studio. Also a few plenary sessions and/or lectures (though less likely in 2021/22).	
Assessment	Project assessment will be based on a combination of: - (~50%) product grade: unique for the whole group, based on both the game itself and the required documentation; - (~45%) process grade (individual), including personal contribution, performance, attitude, and peer evaluation; - (5%) final presentation. The commissioner will be involved both as advisor and as assessor. The final documentation will include writing a scientific paper and actually submitting it to a conference on serious games and/or their application. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr. R. Marroquim	

IN4341	Performance Analysis	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	0/2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	This course applies probability theory and the theory of stochastic processes to the design and performance evaluation of complex networks such as man-made networks as telecommunication, computer and embedded networks and biological networks. The computation with random variables is reviewed. Markov processes and queuing theory will be introduced to the current important concept of "Quality of Service (QoS)" provisioning and to the computation of the blocking probabilities in telephony (both fixed as mobile). Several applications (e.g. the robustness of networks, epidemics in networks, the Internet shortest path routing) are also included. More details are found on brightspace.	
Study Goals	The course intends to provide students with mathematical techniques, in particular probabilistic methods and graph theory, to compare the performance of different network designs and protocols.	
Education Method	Lectures and homework after each class	
Literature and Study Materials	We follow the book Performance Analysis of Complex Networks and Systems, by P. Van Mieghem, Cambridge University Press (2014). See http://www.nas.ewi.tudelft.nl/people/Piet/bookPA.html	
Assessment	Written and closed book. A formularium is provided that can be consulted at the examination. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 3 2021

AP3132	Advanced Digital Image Processing	6
Responsible Instructor	Prof.dr. B. Rieger	
Instructor	Dr. F.M. Vos	
Contact Hours / Week x/x/x/x	0/0/4/2	
Education Period	3 4	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Basics of signal processing, image processing, linear algebra, elementary statistics.	
Course Contents	The course Advanced Digital Image Processing covers the principles of several state-of-art image processing techniques. Particularly, students will study the theory of sophisticated algorithms for: 1. Multi-resolution Image Processing: gaussian scale space, windowed Fourier transform, Gabor filters, multi-resolution systems (pyramids, subband coding and Haar transform), multi-resolution expansions (scaling functions and wavelet functions), wavelet Transforms (Wave series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform (CWT), Fast Wavelet Transform (FWT)); 2. Morpological Image Processing: advanced operations for binary morphology; definitions of gray-scale morphology regarding erosion, dilation, opening, closing; application of gray-scale morphology including smoothing, gradient, second derivatives (top hat) and morphological sieves (granulometry); 3. Image Feature Representation and Description: measurement principles: accuracy vs. precision ; size measurements: area and length (perimeter); shape descriptors of the object outline: form factor, sphericity, eccentricity, curvature signature, bending energy, Fourier descriptors, convex hull, topology; shape descriptors of the gray-scale object: moments, PCA, intensity and density; structure tensor in 2D and 3D: Harris Stephens corner detector, isophote curvature. 4. Motion and optic flow: Taylor expansion method; dual and multi-frame image registration, optic flow; 5. Image Restoration: Noise filtering, Wiener filtering, inverse filtering, geometric transformation, grey value interpolation; 6. Image Segmentation: thresholding, edge and contour detection, data-driven segmentation (boundary detection, region-based segmentation, watersheds, graph-cut, meean shift), model-driven image segmentation (Hough transform, template matching, deformable templates, active contours, ASM/AAM, level sets).	
Study Goals	General learning objectives of the course are: 1. Student has knowledge of can explain the function of state-of-the-art image processing algorithms; 2. Student can solve elementary problems in image processing using Python/MATLAB? programming; 3. Student can solve more advanced problems without implementation, but sketching steps towards a solution; 4. Student can independently acquire new knowledge about image processing from the current literature and present and report about it.	
Education Method	Lectures, practicals and group assignment with plenary presentation and discussion.	
Computer Use	Matlab including the dipimage toolbox and/or other image processing toolboxes.	
Literature and Study Materials	Book 'Digital Image Processing', van R.C. Gonzalez en R.E. Woods, third edition, 2002, ISBN 9780131687288. (Online) Book 'Computer Vision, Algorithms and Applications', R. Szeliski, (http://szeliski.org/Book/). The online version is available for free. We have used the Book Introductory Techniques for 3-D Computer Vision, E. Trucco and A. Verri, ISBN 0-13-261108-2 in the past. Lecture notes Fundamentals of Image Processing (http://homepage.tudelft.nl/e3q6n/education/et4085/sheets/ppt/FIP2.2.pdf) PDF-files of the lecture slides (see Brightspace).	
Assessment	Closed book written exam and assignment. Both parts should be graded 5.8 or higher. A bonus point of 1.5 (to the exam) can be obtained by attending the practicals with 6 out of 8 passed. The final grade is the average of the two parts. The formula for the final grade is: ((0.85*EX + 0.15)+ AS)/ 2 or without the bonus point from the practicals: (EX+AS)/2 With EX the exam grade and AS the grade for the assignment. If you have not passed the exam or the resit, you will need to redo the assignment again next year!	
Permitted Materials during Tests	Closed book exam; books, print-out of pdf files of the lecture slides and lecture notes are not permitted during the written examination.	
Elective	Yes	
Tags	Image processing Matlab Physics	

CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5

Responsible Instructor	Ir. S.E. Verwer
Responsible Instructor	Dr. A. Panichella
Contact Hours / Week x/x/x/x	0/0/4/0
Education Period	3
Start Education	3
Exam Period	3 4
Course Language	English
Course Contents	Software is one of the most complex artifacts of mankind has ever created, but complexity is the enemy of correctness. Modern software testing and validation tools use a multitude of techniques geared toward correct computer code, most of these are base on artificial intelligence. In this course, we study these techniques in details, specifically we will understand and implement: Execution monitoring and taint analysis Branch distance computation Hill-climbing and genetic algorithms Concrete and symbolic (concolic) execution Active state machine learning Genetic programming The goal is to better understand and test software using artificial intelligence. Using the taught techniques you will be able to automatically: Discover which code is reachable Find (security) bugs in software Write tests that cover all reachable code Reverse engineer a code's functionality Patch code to remove bugs and failing tests
Study Goals	The student will: Understand modern AI techniques for software testing. Be able to implement several such techniques from scratch: - smart fuzzing (probing software with input to find crashes/bugs), - symbolic execution (using logic to construct inputs that trigger specific code branches), - fault localization (given that a program fails, find the line of code responsible for the failure), and - automated program repair (using a patch library and genetic programming to improve code) Be able to apply this technology to locate bugs in real-world software implementations.
Education Method	The main part of the course will consist of 3 lab assignments covering the theory (fuzzing&tainting, symbolic execution, automated program repair), and one lab assignment for the application to real software. The students will implement the taught techniques from scratch in the first 3 assignments, which will be scored with a pass/fail. All three assignments need to be passed to complete the course. The final lab will contain a recap from the first three assignments and an application of a state-of-the-art tool on real software. The final lab will be graded and be the final course grade. There will be instruction sessions where students can work on their assignment and ask the teachers for assistance.
Assessment	First three lab assignments (pass/fail). Final lab (100%).
Tags	Artificial intelligence Software

CS4135	Software Verification	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Ir. K.F. Wullaert	
Contact Hours / Week x/x/x/x	0/0/2/0 + 0/0/4/0 practicum	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	How can we ensure that software cannot crash and is guaranteed to be correct? In this course we tackle this question by viewing programs and programming languages as mathematical objects. That way we can use logic to prove properties about programs and thereby guarantee that software is correct. To make reasoning about actual programs and programming languages feasible, we will not be doing these proofs by hand, but instead use a tool called a proof assistant to build proofs that can be checked by a computer. As we will show during this course, proof assistants turn the activity of doing proofs into programming. This course assumes familiarity with functional programming and elementary logic. This course is a specialization course for programming languages and software engineering	
Study Goals	After this course students will be able to: - State and prove properties of functional programs in logic. - Specify the semantics of a programming language in logic. - State and prove the correctness of imperative programs. - Use a proof assistant to perform a mechanized proof.	
Education Method	This course consists of a weekly lecture of 2 hours and a lab session of 4 hours. During the lab sessions students will work on proving simple theorems. Towards the end of the course students will carry out research projects that apply the ideas of the course.	
Literature and Study Materials	Supplementary material: Free online text book "Logic and Proof": https://leanprover.github.io/logic_and_proof/ Free online text book "The Hitchhikers Guide to Logical Verification": https://github.com/blanchette/logical_verification_2021/raw/main/hitchhikers_guide.pdf	
Assessment	The final grade consists of the following parts: - A programming project in a proof assistant. - A written exam Both have weight 50% and both should be 5 or higher. The weighted average should be 5.8 or higher. The research project should be done individually.	
Co-Instructor	E. Visser	

CS4160	Blockchain Engineering	5
Responsible Instructor	Dr.ir. J.A. Pouwelse	
Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	In this course you will learn all aspects of blockchain technology, including tamper-proof data structures, digital identities, transitive trust, fault tolerance, distributed consensus, smart contracts and applications. Ledgers and blockchains are an emerging technology with the potential to radically improve financial transactions, supply-chain flows, transactions in general, and distributed databases. The first three weeks of the course will provide a fast-paced introduction to Bitcoin, Ethereum, and TrustChain developed by TUDelft itself. The main component in this course is a team-based complex engineering project. This course is designed for computer scientists to understand blockchain technology and to produce significant hands-on experience. To provide a deep understanding of blockchain technology and understand why it is special you need to experience first-hand how it operates at a detailed technical level. Students design, implement, and test their own independent project in teams of 3-5 students. Students can choose from a pool of possible project ideas. This course requires you to like software engineering. Topics covered: -Blockchain basics and evolution Bitcoin 1st generation, smart contract generation, future 3rd generation (trust or trust in math) -identity and transitive trust Authentication and security primitives, tamper-proof identities, trust models, MITM attacks, Sybil attacks, and TrustChain by TUDelft -Consensus models Proof-of-work, permissioned, Proof-of-stake, Corda no-global-consensus, TUDelft bottom-up fast consensus model -Smart Contract pro/con encrypted data, Bitcoin scripts, Etherium execution model, Hyperledger + Docker issues, Corda Jar file approach, Tezos difficult to use, powerful technology, vision of the future: trusted verified execution -Markets and exchanges Disruption by open markets, winner-takes-all, and multi-sided market platforms, Uber, Airbnb, 22 years of eBay, Silk Road, honesty among drug dealers, the role of trust in markets, P2P exchange markets -Decentralized Autonomous Organization, novel method to collaborate and organise any economic activity Within this 2021 edition "the Delft DAO" will be prominently featured. TUDelft achieved a world-first in DAO research. We devised a full end-to-end proof-of-principle of a DAO which is capable of 0) near unbounded scalability 1) controlling money 2) democratic decision making and 3) continuous sustained self-evolution. This course provides you with the knowledge to work with this advanced technology. After this course you will have a firm grasp on the current operational blockchain-based systems, realistic view of real-world applications that may be built on top of ledger technology. You will be able to reason and discuss the open challenges and questions that still need to be resolved. This course is a key course for distributed systems students.	
Study Goals	After this course students are able to design and engineer complex blockchain-based systems. Students are able to describe blockchain technology, the various consensus model, smart contracts, markets, and relation to existing database technology. Student are able to setup a new architecture for blockchain applications.	
Education Method	This course consists of four 2-hour lectures. Each lecture is followed by a 4-hour homework period in the same week focused on understanding the background material. In week 1 you will form teams and initiate work on your blockchain engineering project. A list of projects to select from will be provided at the start of this course.	
Literature and Study Materials	Online course textbook: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Narayanan, Bonneau, Felten, Miller and Goldfeder.	
Prerequisites	It is highly recommended to follow this course (see remarks): Security and Cryptography (Q1) Distributed Algorithms (Q2)	
Assessment	The final grade reflects the quality of your work and team cooperation. This course has a minimal amount of formalities. You will write down your project results in a single-page report, IEEE style. You will be graded on your open source efforts located on Github and single-page report. Your grade will be expressed on a scale of 0 to 10. (resits or repair options are not offered for this course) Covid-19 disclaimer: the assessment and course format could be altered at any time !!!	
Remarks	This class has a limited capacity (50). If there is a larger number of enrollments than the capacity of the class, students will be assigned to their preferred blockchain engineering project based on their background, engineering experience level, and match to the course goals. Students who followed Security and Cryptography (Q1) and are also enrolled in Distributed Algorithms (Q2) will have priority for placement. Mathematics students are exempts from this, if they can show some minimal software development experience (e.g. Github profile). Finally, students with a Grade Point Average of 8.0 or higher are eligible for the challenging scientific projects, resulting in a research paper. These project receive intense guidance, but have no capacity limits.	

CS4195	Modeling and Data Analysis in Complex Networks	5
Responsible Instructor	H. Wang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The assignment and final project require basic programming skill.	
Course Contents	Big Data is mostly obtained from features of components and the interactions among components in large complex systems. Examples are (1) end user features and interactions in both online and real-world social networks like Twitter, LinkedIn (2) data from content sharing platforms such as YouTube (3) physiological data of the brain and (4) stock prices etc. in economic systems. Such a dataset is networked in nature i.e. the data of the system components or interactions are (cor)related to each other. This course introduces the basic methodologies to analyze, model, interpret and possibly to predict such Networked Data, combining advances from network science, modeling of dynamic processes and statistical physics, beyond machine learning algorithms. These methods will be applied to diverse real-world datasets obtained from e.g. Facebook, LinkedIn, YouTube, the brain etc.	
Study Goals	After this course, students could construct a network based on the dataset, characterize and model the network in order to e.g. detect patterns and anomalies, model the data via dynamic processes (e.g. viral spreading) on networks to decode the underlying governing mechanisms of e.g. information/error/behavior contagion and to predict e.g. the popularity of a product, news, disease, computer virus, control the contagion process such as maximize the information prevalence and market share. Students could obtain an overview of the Msc/Phd projects on the frontiers of networked data analysis.	
Education Method	In total, there will be about 7 lectures. Students will also learn via an assignment and a final project (each group gets individual supervision).	
Assessment	The final grade is based on the assignment (20%) and final project (80%). There is no resit for both the assignment and the project. disclaimer: information may change depending on the developments around the coronavirus.	

CS4210-A	Algorithms for Intelligent Decision Making	5
Responsible Instructor	Dr. N. Yorke-Smith	
Instructor	Dr. E. Demirovi	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Recommended: IN4010: Artificial Intelligence Techniques, or equivalent; and/or IN4301: Advanced Algorithms, or equivalent Required: basic course(s) in algorithm design and analysis, logic and probability; basic programming (in Python)	
Course Contents	Decision making is at the centre of artificial intelligence. This course gives you practical skills on a solid theoretical base. The course looks at solving mathematical models of NP-hard discrete optimisation problems. These kinds of problems lie at the heart of AI techniques such as planning, machine learning and mechanism design, and more generally combinatorial optimisation. You will learn about a range of modelling techniques from boolean satisfiability to constraint programming, and how advanced solvers for these models work. The course has plenty of real-world case studies as well as theoretical results. Apply the skills you learn in this course by taking CS4210-B: Intelligent Decision Making Project in quarter 4!	
Study Goals	By the end of this course, you will be able to identify features of real-world combinatorial decision problems, and be able to model and design systems for simplified instances of these problems using boolean satisfiability, mixed integer programming, and constraint programming over finite and real domains. You will be able to explain how SAT, CP and LCG solvers work in some detail, and how MIP solvers work at a high level.	
Education Method	Lectures, homework exercises (optional), and programming assignments. The expected workload is: 30% lectures (including preparation for the exams) 40% homework exercises (optional) 30% programming assignments	
Literature and Study Materials	Provided on Brightspace	
Assessment	The final grade depends on the grades obtained for (a) programming assignments (2 in total) [30%] and (b) the exam [70%]. The final grade is computed from the unrounded grades for the components. The final grade for the programming assignment is a uniformally-weighted average of the unrounded grades obtained for the two assignments. Programming assignments can be completed by two students working together. The exam is graded on a scale from 1 to 10. A resit will be available for the exam. The result for the exam is determined by the maximum score obtained for the original exam and the resit. In order to pass the course, the rounded grade (after resit if applicable) for each part of the course must be at least 5.0, and the rounded final grade on the course must be at least 5.8. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Algorithmics Artificial intelligence Group work Modelling Optimalisation Programming Projects Small groups	

CS4225	Educational Technologies	5
Responsible Instructor	Prof. M.M. Specht	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	* Theories of Human Information Processing and Learning * Learning Management Systems * Learning Analytics * Personalisation and Adaptive Educational Systems * Mobile and Seamless Learning Technologies * Artificial Intelligence in Education * Realtime Learning Technologies * Project Design * Project Implementation	
Study Goals	The course will enable you to classify, understand, design and implement the core functionalities and systems for supporting human learning processes. As well current practices implemented as also approaches for technology enhanced learning currently researched will be presented. You will learn how educational technologies provide human learning process support, implement guidance and recommendation, create personalised learning support, as also give real-time feedback and support reflection of learners. In the final project you will identify a problem, design a solution based on the presented approaches and implement your own educational technology solution.	
Education Method	Lectures, weekly assignments and quiz questions, final project	
Assessment	Weekly assignments 30%, Final project 70% disclaimer: information may change depending on the developments around the coronavirus.	

CS4230	Machine Learning 2	5
Responsible Instructor	M. Loog	
Instructor	Dr.ir. J.H. Krijthe	
Instructor	Dr. F.A. Oliehoek	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/4/4	
Education Period	3 4	
Start Education	3	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course is the more advanced and research oriented follow-up to CS4220 [Machine Learning 1]. The content of the latter is, therefore, expected as prior knowledge.	
Course Contents	The course will treat a number of machine learning theories and techniques in detail and on an advanced level. Possible topics : - learning theory - Bayesian networks - online learning - Rademacher complexity - Markov decision processes - semi-supervised learning - reinforcement learning - active learning - causal reasoning and discovery	
Study Goals	After successfully completing the course, the student is able to apply the techniques and theories that have been covered in the course. In addition, they are able to develop learning strategies for new and previously unseen situations. Moreover, the student can provide reasoned justifications for these strategies based, for instance, on theory and/or experiment.	
Education Method	Lectures + Q&A sessions	
Assessment	Grading is based on two parts. Following the lectures -- we have about 11 of those, there is an individual assignment that will be graded pass/fail. In addition, there is a written examination that will be graded on a scale from 1 to 10. You pass the course when you both have a pass for the assignment and a passing grade for the written exam. Upon passing the course, your final grade will be the grade for the exam. Finally, note that there is a resit for the written examination, but not for the report. disclaimer: information may change depending on the developments around the coronavirus.	

CS4235	Socio-Cognitive Engineering	5
Responsible Instructor	Prof.dr. M.A. Neerincx	
Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Basic prior knowledge on human-computer interaction is helpful, but not required.	
Course Contents	Whether you are playing a game in virtual reality, driving a semi-autonomous car, educating yourself in a MOOC, or harmonizing your health and lifestyle via a mobile app; nowadays intelligent networked information and communication technology is omnipresent. This course focuses on the design of human-aware intelligence into such environments, to support joint human-technology performances that bring about positive human experiences (such as social robots that help activity coordinators guide health-promoting games for people with dementia, http://rejam.tudelft.nl). In the Socio-Cognitive Engineering (SCE) course (MSc level), you will become acquainted with the application of a coherent set of methods for the design and evaluation of human--agent collaboration. Based on the SCE-method, we will elaborate on the state of the art of intelligent user interfaces (ePartners), such as artificial personal assistants, artificial team mates, eCoaches, social intelligence, and companion agents. The main topics of study are: - Design methods: Cognitive Engineering, Value Sensitive Design, Scenario-based Design, Claims Analysis, Design Rationale, Design Patterns. - Design for collective intelligence: Knowledge Representation, Ontology Engineering, Mental Models, Theory of Mind, ePartners, Adaptive Automation, Socially Intelligent Interfaces. - Design Evaluation: Prototyping, Test Methods, Measures, Questionnaires, Ethics. - Human Factors Theories and Models: Human Cognition & Learning, Memory, Emotion, Task Load, Human-Agent Teamwork, Behavior Change and Persuasive Technology.	
Study Goals	At the end of this course, students will be able to: 1. Explain the essential concepts of the design methods addressed in the course. 2. Explain the (dis)advantages of various design methods and their complementarity. 3. Apply the design methods addressed in the course in their research and design projects. 4. Explain what a design rationale is. 5. Construct a design rationale. 6. Create design specifications that are grounded in a design rationale. 7. Evaluate the strengths and weaknesses of a design rationale, e.g. using human-centered evaluations that test the design rationale. 8. Explain some of the state of the art human factors theories, models, and methods relevant to intelligent user interfaces, human-agent collaboration, and ePartner technology. 9. Write a structured report about a design-test cycle, with sufficient detail for a new group of researchers to continue the research. 10. Present work on a design project to an academic audience. 11. Work in a group on collaborative assignments.	
Education Method	LECTURES During the lectures, the teachers will present a range of theories, models, and methods relevant to socio-cognitive engineering. Students are required to read a number of scientific papers which are made available on Brightspace, along with the sheets/slides of the lectures. Together, the sheets/slides and the papers provide the students with the required theoretical knowledge to work on the practical project, and to learn about relevant design methods, human factors theories, conceptual solutions, and design principles. Most of the lectures include practical assignments and discussions stimulating the students to apply the contents of the lecture to their project (also see Project). PROJECT In the project, students work in groups to apply the knowledge acquired during the lectures. Students are required to plan, execute, present, and report on a complete design cycle (i.e. design, prototype, and evaluation) for a given design problem. This year (like the past years), the design problem is a social robot for older adults with dementia, and their social environment (https://rejam.tudelft.nl). The objective of the social robot is to improve humans physical, social, cognitive, and emotional well-being. The students will use the Wiki Socio-Cognitive Engineering (WiSCE) tool to specify the design rationale and its evaluation, step-by-step (see also https://scetool.ewi.tudelft.nl). Throughout the course, students will give presentations about their progress, on the design and evaluation of their prototype.	
Literature and Study Materials	Papers from scientific journals on Brightspace. Lecture notes on Brightspace.	
Assessment	Literature and study material consist of: - Papers from scientific journals on Brightspace. - Lecture notes on Brightspace The module assessment concerns the processing and application of the theory and methods; the construction of the design (rationale) and the evaluation; and the provision of the resulting concise and coherent report (including the lessons learned): Presentations (10%) Prototype (10%) Project report according to the prescribed format (70%) Individual reflection (10%)	
Exam Hours	There is no exam. The assessment is based on a paper, presentation and report. During the course, students will receive feedback on interim work. There is no resit after the end of the course.	

CS4240	Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	Seminar Computer Vision by Deep Learning	
Expected prior knowledge	Basic pattern Recognition, Basic Machine learning, Basic statistics, Basic probability theory, Programming experience (python + numpy). Note: Without some affinity/experience with python (and numpy) programming it might still be possible to pass the course, but will become quite difficult.	
Course Contents	In this course we will look at a specific field of Artificial Intelligence and Machine Learning: Deep learning. Deep learning has shown remarkable success with large data sets and unstructured input data such as raw images/audio/text. Topics include: feed forward networks, back-propagation, optimization, convolutional nets, recurrent nets, self-attention, unsupervised methods. The course will have lectures, a seminar, a lab practical and a project: - The lectures will be on generic topics, following the book; building the backbone. - The lab assignments will have you apply basic concepts of the lecture in python notebooks. - The seminar will have students read, critique, and present relevant deep learning research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The project will have students apply and design their own (small) deep learning project in the context of scientific reproduction.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the different deep learning techniques reviewed in the course, such as SGD, MLPs, CNNs, RNNs, GANs. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [LO4]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (eg: Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) problem domain which may need to re-implement, run, evaluate, investigate, extend existing research or code [LO8] Write clearly and concisely about your code, method, results, and analysis.	
Education Method	Lectures for basic theory based on the literature Assignments: we help you to become familiar with PyTorch; applying concepts from the lecture on small problems. Lab project: design and execute your own deep learning project in the context of scientific reproductions using https://reproducedpapers.org/ Seminar: paper reading, critiquing, and presenting.	
Literature and Study Materials	Books: freely available online: - http://www.deeplearningbook.org/ - https://d2l.ai/ Research papers will be made available through Brightspace. Assignments are based on PyTorch: https://pytorch.org/	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Project: in a small group of students you work on a deep learning paper reproducibility project. 3. Exam about the papers, assignments, and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4275	Web Programming Languages	5
Responsible Instructor	E. Visser	
Instructor	Dr.ir. D.M. Groenewegen	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	Languages and frameworks for web programming are constantly evolving. Over the past decade, there has been a large shift from applications with server-side rendering of separate web pages, to single page applications with client-side rendering and web service back-ends. One of the strengths of web programming technologies is separation of concerns. The concerns such as describing content, styling, behavior, and persistence, are often separated with their own domain-specific languages. A particular programming problem that newer web programming languages tackle is dynamic user interfaces with automatic fine-grained updates. This problem is not restricted to web applications, but applies to any GUI programming abstraction. Consequently, the technologies for web programming are also relevant for development of cross-platform mobile and desktop applications. In this course, students will analyze web languages and frameworks from a programming language perspective. They will explore the underlying concepts and abstractions, and learn how the tools relate to each other. The investigated web technologies range from more traditional full-stack web development solutions with persistence and templating, to popular client-side UI solutions with fine-grained updates and state synchronization.	
Study Goals	The course gives students the conceptual and technical skills to understand the role of programming languages in web programming and the advantages and disadvantages of different approaches.	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab sessions - reading lecture material and papers - making project assignments	
Assessment	Students get a grade for each of the project assignments. The final grade is the weighted average of the grades for the project assignments. There will not be a resit for the course. Disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is the average of the grades for the project assignments.	

CS4400	Deep Reinforcement Learning	5
Responsible Instructor	Dr. J.W. Böhmer	
Instructor	Prof.dr. M.T.J. Spaan	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students must have passed IN4010(-12) "Artificial Intelligence Techniques", or have acquired equivalent knowledge about: - basic probability theory, analysis and algebra - general machine learning methodology, e.g. regression - fully and partially observable Markov decision processes - tabular reinforcement learning methods, e.g. Q-learning - the exploration/exploitation trade-off, e.g. RMAX or UCB - multi-agent learning, e.g. centralized training and decentralized execution Students are encouraged to close any gaps in the above knowledge and to familiarize themselves with the Python/PyTorch deep-learning framework before the start of the course.	
Course Contents	This course will cover the breadth of modern model-free RL methods, discuss their limitations and introduce a variety of current research topics. In particular, we expect to cover the following: - deep learning methodology and architectures - stabilization of approximated value estimation - modern actor-critic methods - planning as inference - exploration with deep networks - offline reinforcement learning - deep multi-agent reinforcement learning - multi-task and meta learning	
Study Goals	After successful completion of this course, students - can list the strengths and limitations of modern deep RL approaches, - explain the underlying concepts of the discussed methods, and how they differ from each other, - can implement selected algorithms/architectures, and - can analyze a new task to decide which algorithms/architectures to apply.	
Education Method	The course will be taught in one lecture per week and the content will be solidified in homework, which will be presented in one mandatory tutorial per week.	
Assessment	The final grade will be 100% determined by a written exam at the end of Q3, with a resit possibility in Q4. To be eligible for the exam, students must attend weekly tutorials and hand in homework exercises. Homework will not be individually graded, but at least 75% of the answers must be of sufficient quality (in terms of time commitment, not necessarily correctness) to be eligible to take the exam.	
Maximum number of participants	As this is the first time the course will be taught, it will be restricted to 30 participants.	

CS4405	Analysis of Concurrent and Distributed Programs	5
Responsible Instructor	Dr. B. Özkan	
Responsible Instructor	Dr. S.S. Chakraborty	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software systems are becoming highly concurrent and distributed to utilize modern multicore architectures and increasing speed and bandwidth in networks. Shared-memory concurrency in multicore programs and message-passing concurrency in distributed programs share many common abstractions and problems. In the multicore era, all performance-critical software employs some form of concurrent programming; typically shared memory concurrency. In this setting, programmers use a number of primitives to develop efficient and correct concurrent programs. To do so the programmers have to understand the behaviors of the primitives and reason about them. It is also important to match the programming paradigms and underlying architectures. For instance, traditionally programmers have assumed that a multithreaded program executed simply by interleaving the executions of its threadsa model known as sequential consistency (SC). This assumption is, however, invalidated both by mainstream multicore architectures, which often execute instructions out of order, and by compilers, whose optimizations affect the outcomes of concurrent programs. As a result, concurrent programs have more outcomes than SC allows. In the distributed setting, the units of concurrency are independent processes that do not share memory but communicate by exchanging asynchronous messages. The execution of such a system involves two main sources of nondeterminism: concurrency and partial failures. As the processes run concurrently, the exchanged messages can be delivered and processed in many different orderings. The distributed set of processes is also prone to network of process failures. The trade-off between the systems availability in the existence of failures and the consistency between the processes gives rise to a spectrum of weak consistency notions. It is important to reason about concurrency, possible failures, and consistency guarantees to implement distributed programs correctly and understand their behavior. This course aims to explore analysis techniques for concurrent and distributed programs. Outline of Lectures: Shared memory concurrency: - Abstractions for shared memory concurrency - Relaxed memory concurrency - Correctness of concurrent programs Distributed concurrency: - Distributed system components, models and assumptions - Fundamental abstractions for distributed systems	
Study Goals	This course aims to give students a deep understanding of concurrency and distribution in modern systems and hands-on experience for analyzing these systems. At the end of the course, the students will be able to: - Analyze and reason about concurrent and distributed programs - Apply and analyze existing techniques on unseen problems - Be able to pursue independent further research in the area	
Education Method	The course consists of the following education methods: - Lectures for reviewing concurrency and distribution concepts - Homeworks/assignments - Developing a course project, writing a report, and presenting it (course project) To finish the course, students (in teams) will have to: - Study several papers which will be discussed during the lectures - Deliver their assignments - Deliver and present their implementation project	
Assessment	The final grade is composed of: research project implementation) (40%) + research project report (20%) + research project presentation (20%) + homework assignment (10% + 10%). No written exam. Resits are not offered.	

CS4415	Sustainable Software Engineering	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	L. Miranda da Cruz	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	Sustainable Software Engineering is an overarching discipline that addresses the long-term consequences of designing, building, and releasing a software project. By definition, sustainability covers five main perspectives: environmental, social, individual, economic, technical. This course mainly focuses on the first, also known as Green Software Engineering. Incidentally, we will also cover some fundamental aspects of social and individual sustainability of software projects. Software Engineering (SE) has long addressed sustainability by narrowing it down to economic and technical sustainability. However, our society is facing major sustainability challenges that can no longer be overlooked by software engineers and computer scientists. It was estimated that, by 2040, the ICT sector will contribute to 14% of the global carbon footprint. Hence, environmental, social, and individual ought to be part of the equation when it comes to design, build, and release software systems. The problem is far from simple, but we need expert computer scientists to bring sustainability into the core values of the next generation of tech-leading organizations.	
Study Goals	After attending this course, you will be able to: LO1. Measure software energy consumption. LO2. Automate carbon-awareness in software development. LO3. Discuss sustainability principles. LO4. Solve sustainability issues in real software projects. LO5. Propose innovative strategies to monitor software sustainability.	
Education Method	To meet these objectives, you will be involved in a broad set of learning activities: lectures, paper reading, software analysis, software development, essay writing and presentation. These heterogenous set of activities aims at building a strong set of hard skills for energy-efficient code development combined with a strong set of soft-skills and critical thinking. Ideally, you will work on projects that will also help real-world software projects embrace a green software culture.	
Assessment	The assessment will be performed as part of the group project. It will include several steering meetings, an essay, a software repository, and a final presentation.	

CS4430	Network Security	5
Responsible Instructor	Dr.ing. A. Zarras	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	The course provides an overview of the most important concepts, methods, and best practices in computer and network security. In this course, students will obtain the knowledge and hands-on experience to secure networking and communication systems. The course's primary focus will be on technologies, protocols, attacks, and defenses. More precisely, starting from a review of common vulnerabilities and attack scenarios, the course will discuss the fundamentals of security engineering and their application in system design, review tools and methods to assess and test communication infrastructure from a security perspective. As a result, students will gain theoretical knowledge and hands-on experience in network attacks and defense methods. Knowledge activation and the transfer from conceptual understanding towards practical experience will be further facilitated by students implementing their own attack or defense tools on selected topics, as well as conducting measurements on the effectiveness of attack and defense schemes.	
Study Goals	See course contents.	
Education Method	Lectures, Labs, and Project.	
Assessment	Assignments and Project.	

EE4560	Information Theory	5
Responsible Instructor	Dr. J.A. Martinez Castaneda	
Instructor	G. Joseph	
Instructor	Dr.ir. J.H. Weber	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	This course explains the basic ideas of information theory and the correspondences between the elements of this theory and certain natural concepts of importance in a wide number of fields, such as transmission, storage, authoring and protection of data. On the basis of simple concepts from probabiliby calculus, models are developed for a discrete information source and a discrete communication channel. Further, the theoretical basics for developing source coding algorithms is provided, as well as the basics of optimal data transmission through a discrete communication channel. The following topics will be covered: * (Differential) Entropy, Relative Entropy and Mutual Information * Asymptotic Equipartition Property * Data Compression * Channel Capacity * Gaussian Channel * Rate-Distortion Theory * Network Information Theory	
Study Goals	Upon completion of this course the student will understand the fundamentals of Information Theory, which includes the following: (a) the correspondences between the elements of this theory and certain natural concepts of importance in a wide number of fields, such as transmission, storage, authoring and protection of data, (b) core theorems of information theory, (c) the models that are developed for a discrete information source and a discrete communication channel on the basis of simple concepts from probability calculus, (d) how to develop source coding algorithms, and (e) how to secure optimal data transmission through a (noisy) discrete communication channel.	
Education Method	lectures + mini project	
Assessment	CoVid-19 disclaimer: In light of the Corona crisis a remote assessment format could be implemented. Examination: Project and Exam The grade is determined by a project score (20%) and an exam score (80%). There are two opportunities for both. These are further explained below. Please note the exam format will depend on current CoViD-19 regulations. Project: The project is individual. Detailed instructions are listed in Brightspace, the project report is to be delivered via Brightspace. Exam: If regulation allows standard written examination on campus will be applied, otherwise we will use YouSeeU (Virtual Classroom) which is embedded in Brightspace. All details regarding the examination are listed in Brightspace. Grading: First opportunity: The project report should be submitted before the deadline (listed in Brightspace). The project score P1 is an integer between 0 and 10, while the exam score E1 is between 1 and 10 with a half-integer accuracy. The total weighted score is then (4×E1+P1)/5, which is rounded to the nearest grade in the set {5.0, 5.5, 6.0, 6.5, , 9.0, 9.5, 10.0} if both E1 and P1 are at least equal to 5. In case one or both are below 5, then the total weighted score is min(5.5,(4×E1+P1)/5), which is rounded to the nearest grade in the set {1.0, 1.5, 2.0, 2.5, , 5.0, 5.5}. In other words, a necessary condition to pass the course is that both the project score and the exam score must be at least equal to five. N.B.1: If the project report is not sent before the deadline, then P1=0. N.B.2: If the student does not participate in the exam, then E1=0. N.B.3: If the student already did an ee4560 project in a previous study year, then the student can request one of the lecturers before the project deadline by e-mail to let this be taken into account; this can be done in two ways: either to let the score of that project count as P1, or to take a different project from the list and to submit the report before the deadline, in which case P1 will be the maximum of the old and the new score. Second opportunity: A student not passing in the first opportunity or willing to improve his/her grade can redo the project, redo the exam, or both. In case a new project is done, the topic should be different. The project report should be submitted before the resit deadline (listed on Brightspace). A resit exam will be announced in Brightspace. With the project and/or exam scores being P2 and E2, the new total weighted score becomes (max{P1,P2}+4×max{E1,E2})/5, which is rounded as indicated above, still requiring that both the project score max{P1,P2}and the exam score max{E1,E2} must be at least equal to five. disclaimer: information may change depending on the developments around the coronavirus.	

ET4394	Wireless IoT and Local Area Networks	5
Responsible Instructor	Dr. P. Pawelczak	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Students are advised to follow the course Wireless Communications (ET4358) before taking this Wireless Networking course. An advantage is to have entry-level programming skills (Matlab, Python, C/C++). Nonetheless, students with little knowledge of programming will be helped.	
Course Contents	DISCLAIMER: this study guide information may change depending on the developments around the corona virus. The following modules will be discussed during the lectures: Introduction (example topics): - What is wireless networking - Where to search for (academic) wireless network literature and resources Medium Access Control (example topics): - WiFi: hidden/exposed terminal problem, Carrier Sense Multiple Access - Bluetooth standard: in-depth look into the channel hopping, protocol specifications WiFi (example topics): - Review of IEEE 802.11 standards - Protocol format - ISM band regulation - Adaptive Modulation and Coding - WiFi Matlab class (assignment) IoT networking standards (example topics): - LoRa: protocol specifications, energy consumption, modulation format, network design Review of wireless tools (example topics): - Introduction to wireless packet sniffing and analysis using Wireshark (assignment) - Simple simulations of WiFi network with NS3 RFID networking (example topics): - Principles of backscatter - Protocol formats: EPC C1G2 - RFID hackathon (assignment) Cognitive radio (example topics): - Basics of spectrum management - White Space Databases - Theory of spectrum sensing	
Study Goals	At the end of the course students will be able to: (i) to understand how practical wireless systems work and get a deeper understanding of how the theoretical concepts of wireless communications apply to practice; (ii) employ their own analysis methodology to assess new wireless network systems (especially at the physical layer); (iii) understand rapid prototyping of new wireless systems (for instance, with software defined radio).	
Education Method	Lecture presentations, mini-project assignments, assigned paper reading and its critical analysis and presentation.	
Computer Use	Each student should have its own laptop (preferably with a Linux distribution, where Linux must not be installed on a virtual machine). We will be using Matlab, and/or NS3 and/or GNURadio and/or Wireshark for the assignments.	
Books	WiFi Matlab WLAN toolbox: https://nl.mathworks.com/help/wlan/; Wireshark learn page: https://www.wireshark.org/#learnWS; tutorial on NS3 network simulator: https://www.nsnam.org/documentation/; specific chapters from books provided at the beginning of each lecture.	
Prerequisites	Background in programming (Matlab, Python, Bash)	
Assessment	Points from the mini-project assignments. A research paper analysis from conferences such as IEEE INFOCOM, ACM MobiCom, ACM SIGCOMM will be required to pass the course. disclaimer: information may change depending on the developments around the coronavirus.	

IN4152	3D Computer Graphics and Animation	5
Responsible Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. R. Marroquim	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students that haven't followed any previous Computer Graphics courses (like TI1806) will be able to participate, but might have to invest some more time to catch up in the first lectures.	
Course Contents	Have you ever wondered how Toy Story was made, why the game Last of Us 2 looks so beautiful, or have you ever wanted to create your own graphics application or game? Then you should consider following this course. If not, then you should still follow it... maybe, you will become interested! In this course, you will get a good idea of Computer Graphics in general. The topic is of very high relevance for the industry and the research community and has numerous applications in different domains, such as scientific visualization, video games, simulators, special effects, animated movies and many more. Here, you will learn about basic algorithms, as well as modern techniques. We will address several topics: the principles of image synthesis, object representations, geometric and hierarchical transformations, graphics cards and the graphics pipeline, realistic rendering (including global illumination and effects, such as reflections), expressive rendering, physics simulations, rendering control (including previsualization systems used by professionals in the movie industry), and perceptual rendering, which relies on properties of the human visual system to enhance the quality of the images. Besides course sessions on the theory of Computer Graphics, some of the algorithms will also be reproduced in practice, and deepened during the final project.	
Study Goals	The course teaches computer graphics techniques on an advanced level. After the course the student is able to classify the different modeling, shading, and display techniques. The student can reproduce the basic mathematical and algorithmic notions associated with these concepts, can comment on the weak and strong points of these techniques, and can apply the core concepts within a graphics program in practice.	
Education Method	lectures, instructions, research papers, lab work	
Literature and Study Materials	Research Papers in domain of selected topics, lecture sheets, online sources, optional books (see below)	
Books	Fundamentals of Computer Graphics by Shirley et al. - CRC Press Real-time Rendering by Tomas Akenine-Möller, Eric Haines, Naty Hoffman - Peters, Wellesley Real-Time Shadows by Elmar Eisemann, Michael Schwarz, Ulf Assarsson, Michael Wimmer - Taylor & Francis Computer Graphics. Principles and Practice by James D. Foley, Andries VanDam, Steven K. Feiner - Addison Wesley	
Assessment	The course will be evaluated with two grades, a project grade, accounting for 60%, and a paper grade 40%. The project grade is the result of a project and its presentation that is building upon the assignments that are handed out (roughly) weekly during the duration of this course. The paper grade is the result of the presentation of a scientific paper and the development of an associated practical implementation. Details of both elements will be presented during the lecture. Both grades (project and paper) have to be at least a 5.0 and their weighted average should be 6.0 or higher after rounding (0.5 steps).	

IN4253ET	"Hacking Lab"-Applied Security Analysis	5
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Necessary background differs per student project, see first lecture or contact instructors for details	
Course Contents	The security of computer and telecommunication systems is becoming an increasing concern. In this course, we will review the current state of the art on security research and gain practical experience in assessing the security and vulnerabilities of communication systems. Engineers are typically taught to focus on performance, correctness, scalability, and maintainability when building communication and information processing systems. However, an additional set of design principles are required to achieve security. In this course, we discuss security principles, common pitfalls and vulnerabilities. The weekly lectures provide an introduction into security research, with a focus on real-world security, privacy-enhancing technology and common security pitfalls. Each student participates in a "Hack Project", with a group of one to four students. Students can select between a wide range of available Hack Project outlines within the first week. The goal may be to evaluate the security of a real-world IT system, developing a proof-of-concept exposing a vulnerability or focussed on preserving privacy in a post-Snowden world. Students may propose their own Hack Project based on their background knowledge and skills. Such Hack Projects need to be approved and shaped together with the instructor. Example of possible outlined hardware-oriented projects are: development of a wifi tracker, programing an FPGA system to break passwords, assess the security of RFID cards, or to transparently intercept Ethernet traffic. Concrete software projects are: hacking Bitcoin, improving the TOR anonymity protocol and create Android-based tools for human rights activists in Iran, Egypt and Russia, reprogramming neural networks attacks. Each Hack Project is documented with a written report. This can be in the form of a 6-8 page IEEE-style scientific article or a traditional more lengthy report. All results, experiences and findings are presented to the entire class in the last week of the course. Hack Projects also report their progress several times during the course, after the weekly lectures.	
Study Goals	After this course, the student will have a thorough knowledge of security in real-world systems, and will be able to explore the literature on this topic independently. The student will be aware of the poor state of security in real-world computer systems. The student can explain the common pitfalls, why these known failures still occur and reasons behind the poor state of security in general.	
Education Method	Lectures, student presentations, written final report and active participation. Attendance and active participation during lectures is mandatory. This sadly means telelecturing is not possible.	
Literature and Study Materials	Customize literature lists and study materials are provided per project topic	
Assessment	The final class grade is composed of several partial grades. Partial grades are given for the written Hack Project report (60%), final presentation of result (10%), presentation of ongoing project progress (20%), participation in discussions, overall quality of the practical work and class attendance (10%). Students are required to obtain a passing grade on all partial grades. Attendance to lectures is mandatory. No final written exam. No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	If there is an unexpected high demand for this course, then enrollment will be based on past performance in relevant courses.	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4325	Information Retrieval	5
Responsible Instructor	Dr. C. Hauff	
Responsible Instructor	J. Yang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.	
Course Contents	Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs. Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines. Covered topics include: = Information Retrieval Models; = Indexing Techniques; = Web Search; = Information Seeking Paradigms; = Evaluation of information retrieval systems; = Components of Natural Language Processing; = Word Embedding Techniques; = Neural Language Models; = Deep Contextual Embeddings; = Neural Information Retrieval	
Study Goals	At the completion of this course, students will be able to: = Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1] = Describe and implement different indexing techniques. [Learning Objective 2] = Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3] = Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4] = Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5] = Describe typical NLP tasks and components of an NLP system. [Learning Objective 6] = Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7] = Compare word embedding and contextual embedding techniques. [Learning Objective 8] = Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]	
Education Method	Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments. Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.	
Literature and Study Materials	Scientific papers, course slides, course books - all resources are available on Brightspace.	
Books	Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA. Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press. Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson. Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing. Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".	
Assessment	Weekly individual assignment, weighting 10% of the final grade. Two group projects for IR and NLP, each weighting 45% of the final grade. The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities. disclaimer: information may change depending on the developments around the coronavirus.	

IN4343	Real-time Systems	5
Responsible Instructor	Dr. G. Iosifidis	
Instructor	Prof.dr. K.G. Langendoen	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures & 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	3TU MSc Embedded Systems; the corresponding courses are 2IN26 at TU Eindhoven, and 312030 at TU Twente	
Expected prior knowledge	Basic software engineering, C system programming, basic Linux operating system knowledge	
Course Contents	- basic concepts of RTS - worst case execution time estimation - scheduling policies - response-time analysis - jitter analysis - handling overload - multiprocessor scheduling - reservation-based scheduling	
Study Goals	The course intends to bring the student into the position to: - Explain the fundamental concepts and terminology of real-time systems - Construct task schedules using different scheduling policies under a given set of realistic system constraints - Analyze the timing behavior of a system for a given system model and scheduling policy - Discuss advantages and disadvantages of different scheduling policies for a given platform or system - Discuss the effect of hardware and software interferences on the timing behavior of a given system - Identify (reverse engineer) parameters of a scheduling scheme or a task set from output traces of the system - Derive (reverse engineer) the system specification from a given implementation (in the lab) - Evaluate the scheduling overheads of a given implementation (in the lab) - Implement event-based scheduling policies on a given microcontroller (in the lab)	
Education Method	lectures with exercises (32 hrs); self study (78 hrs); lab assignments (30 hrs)	
Books	Hard Real-Time Computing Systems by G.C. Buttazzo, Springer 2011	
Assessment	Written exam (grade) + lab work; the exam has a resit disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	3	
Permitted Materials during Tests	Simple calculator	

IN4391	Distributed Systems	5
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Starting with the mid-1990s, computing is undergoing a revolution, in which collections of independent computers appear to users as a single, albeit distributed, computing system. Motivated by the increase in the computation capacity of consumer computers, by the commoditization of server-grade machines, and by the advent of the Internet, the distributed computing paradigm has permeated all fields using computers. Current distributed computing applications range from the consumer social networks, peer-to-peer file-sharing, and massively multiplayer online games; to scientific computing using Big Data and distributed sensors; and to engineering fields and industrial control systems. This course focuses on the systems aspects of distributed computing. Specific, contemporary distributed systems are used as illustrative examples to discuss system design and non-functional requirements.	
Study Goals	1. Explain the objectives and functions of distributed computing systems. 2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems. 3. Describe the architecture and operation of distributed computing systems. 4. Explain how distributed computing systems can process user workloads. 5. Explain how distributed computing systems can detect and correct faults and errors. 6. Implement complex operations of modern distributed computing systems in realistic scenarios. 7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)	
Education Method	Lectures: This class uses a partially flipped classroom setting. Students are required to prepare parts of the material through self-study to be able to follow the in-class discussion. This involves, e.g., reading scientific papers. Practical: Designing, implementing, and evaluating a complete distributed system in groups, based on existing research work. Multiple topics are given to choose from. Deliverables include the code and a report of max. 10 pages.	
Literature and Study Materials	Textbook: Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Principles and Paradigms (2nd Edition), Prentice Hall, 2006. The textbook introduces the student to the traditional theory of distributed systems. Additional material: Several relevant research articles introduce the student to the classic literature as well as the latest advances on the topic.	
Assessment	Written exam (closed-book, open questions) during the exam period. A resit for the exam is offered in the following exam period. Practical project assessed based on the code, a presentation, and the report. This course uses gamification. Points can be collected through the practical project (max 4000 points) and the final exam (max 6000 points). The final grade is determined proportional to the 10000 points total. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Specialistievakken start kwartaal 4 2021

CS4035	Cyber Data Analytics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	The course provides theoretical and practical background for applying data analytics in the field of cyber security. Cyber data analytics is a huge field with a great diversity of techniques and applications. The course is centered on a selection of seven such techniques: learning from imbalanced data; behavioral profiling and anomaly detection; sequential data mining; data stream mining; learning from software data; adversarial machine learning; and privacy-aware data mining; Anomaly detection is one of the main topics in cyber security. Specific difficulties that the student will learn to handle are the huge amounts of data and the large number of false positives. Behavioral profiling applies to both people and software processes. Different techniques will be taught to construct profiles from software logs. While building such profiles, care should be taken to not infringe upon the privacy of individuals the data is collected from. Finally, attackers will modify their behavior in order to avoid being detected, a cyber data analytics engineer tries to make their models/profiles robust against such modifications.	
Study Goals	The student will be able to: Apply machine learning to real data Understand and modify machine learning algorithms Learn models from time series Detect anomalies in multidimensional time-series Use distributed processing to speed up machine learning Learn models from data streams with limited memory Learn sequential models Use machine learning for fingerprinting and profiling Preserve the privacy of data owners while learning models Learn robust models that can detect evasive attackers Use machine learning to detect fraud, attacks, and botnets	
Education Method	There will be two lectures for each of the seven topics, and 3 large lab assignments on fraud detection, anomaly detection, and behavioral profiling, and 1 smaller lab on adversarial robustness. There is no exam. Teams of two students will work on these assignments which contain both individual and collaborative components. Deadlines are strict as peer-review will be used to both learn of other possible solutions, provide feedback, and get initial estimates on the obtained grade.	
Assessment	3 large lab assignments on fraud detection (30%), anomaly detection (30%), and behavioral profiling (30%), and 1 smaller lab on adversarial robustness (10%).	

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4125	Seminar Research Methodology for Data Science	5

Responsible Instructor: Dr.ir. W.P. Brinkman

Instructor: Dr. K.A. Hildebrandt

Instructor: J. Urbano Merino

Contact Hours / Week x/x/x/x: 0/0/0/4

Education Period: 4

Start Education: 4

Exam Period: 4
5

Course Language: English

Expected prior knowledge: basic knowledge in mathematics (linear algebra, calculus, probability and statistics)

Course Contents

The course focuses on research methods for data science. It looks at underlying principles and concepts for data collection, analysis and data processing, as well as the use of tools to do this.

The main topics of study are:
Conceptualizing research questions and experimental design
Frequentist and Bayesian data analysis
Generalized linear models for statistical analysis
Multilevel modelling for hierarchical and longitudinal data analysis
Measuring and sampling, validity and reliability
Linear and nonlinear dimensional reduction
Principles of statistical testing

In the course, students will be using software tools such as R, and Matlab/Mathematica

Study Goals

The main aims of this module for the student is to achieve understanding of research methods for data science and obtain practical experience with data analysis and data processing methods. This module provides students with the opportunity to develop and demonstrate their understanding, knowledge, and competence. The learning outcomes for the module are that students will be able to:
1. Appreciate and comprehend strategies for collecting and processing data to answer data-driven research questions
2. Understand and reproduce key principles underlying statistical data and data processing analysis
3. Learn to identify and avoid typical biases, paradoxes and misunderstandings in data-driven research
4. Apply and select appropriate data modelling techniques to analyse data and data processing

Education Method

Lectures/Assignments

Expected Workload

Lectures: 26 hours (13 × 2 hours lectures)
Reading time: 39 hours
Preparation basis tool use: 25 hours (5 × 5 hours for each tool)
Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours)

Total = 140 hours

Literature and Study Materials: Will be provided online

Assessment

Course will be assessed on 3 coursework assignments.
A) Analysis of experimental research data (40%)
B) Exploration of real-world data set (20%)
C) Linear and nonlinear dimensional reduction (40%)
Students work in small groups on the 3 assignments. For each assignment, the student group submit a report and give a presentation including a question and answer round where individual group members are assessed on the coursework. The final course mark is the weighted average of the three assignment marks. Note that, there is a minimum grade of 5.0 for each assignment grade and an average grade for all components of at least a 5.8 in order to pass the course. Also, marks for individual assignments do not carry to the next year.

Resit next quarter
Resubmission of modified coursework is only allowed for assignments that received a fail mark (<5.0). Overall resit mark will be capped to 6.0.

disclaimer: information may change depending on the developments around the coronavirus.

Permitted Materials during Tests: NA

CS4140ES	Embedded Systems Laboratory	5
Responsible Instructor	Prof.dr. K.G. Langendoen	
Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/4 Lectures + 0/0/0/4 Lab	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	MUST have C programming skills. Students who have taken the CSE2425 Emb. Software course automatically qualify, others will have to pass an on-line ACCEPTANCE test.	
Course Contents	This highly multi-disciplinary course comes with a lab project where teams of 4 students each will have to develop an embedded control unit for a tethered electrical model quad rotor aerial vehicle (the Quadrupel drone), in order to provide stabilization such that it can hover and (ideally!) fly, with only limited user control (one joystick). The control algorithm (which is given) must be mapped onto a home-brew PCB holding a modern RF SoC interfacing a sensor module and the motor controllers. The students will be exposed to simple physics, signal processing, sensors (gyros, accelerometers), actuators (motors, servos), basic control principles, and, of course, embedded software (C) which is the programming language to be used in order to develop the control system. The project work (including written report) covers the entire duration of the course period, and will take approximately 128 hours, of which 32 hours are spent at the lab facilities. This is a core course of the Masters in Embedded Systems.	
Study Goals	Student is acquainted with real-time programming in an embedded context, along with a basic understanding of embedded systems, real-time communication, sensor data processing, actuator control, control theory, and simulation. Moreover, the student has had exposure to integrating the various multidisciplinary aspects at the system level.	
Education Method	Lectures (8*2hrs), lab work (8*4hrs), coding@home (8*12hrs), report (8hrs), so on average 2 days per week	
Literature and Study Materials	Lecture notes + Website	
Assessment	Lab. project (75%) + written report (25%), no exam, no resit disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	The capacity is limited and -as this is a compulsory course for ES students- they get preference over other MSc students.	

CS4145	Crowd Computing	5

Responsible Instructor Prof.dr.ir. A. Bozzon

Responsible Instructor U.K. Gadiraju

Contact Hours / Week x/x/x/x 0/0/0/4

Education Period 4

Start Education 4

Exam Period none

Course Language English

Expected prior knowledge Basic knowledge of artificial intelligence and/or human computer interaction is advised. Proficiency in at least one programming language.

Course Contents

Crowd Computing is an emerging field that sits at the intersection of computer science and data science. Crowd computing studies how large groups of people can solve complex tasks that are currently beyond the capabilities of artificial intelligence algorithms, and that cannot be solved by a single person alone.
It involves the algorithmic engagement and coordination of people by means of Web-enabled platforms. These complex tasks are mainly focused on the creation, enrichment, and interpretation of data, making crowd computing a building block of data science. Examples of such tasks include the coordinated creation of data about real world events when electronic sensors are not available; the annotation of existing data sets to create ground truth data for the training of machine learning algorithms; and the analysis and interpretation of Web data to spot identify inappropriate content (e.g.,hate speech, or fake news).
Crowd computing is an essential tool for any data-driven company: from Facebook to Microsoft, from Google to IBM, from Spotify to Pandora, all major companies employ crowd computing to fulfil their data needs, both by involving employees, and by reaching out to anonymous crowds through online marketplaces like Amazon Mechanical Turk or Appen. Crowd computing methods therefore play an important role in the design, development and evaluation of a variety of products, services, and systems in a variety of domains.

The objective of the Crowd Computing course is to introduce the scientific and technical underpinnings of crowd computing, and to investigate how it can be used for computer science applications (e.g., information retrieval, machine learning, next-generation interfaces, and data mining) and for real world applications (e.g., cultural heritage preservation, online knowledge creation, smart cities, etc.)

The course is designed around one key challenge, the creation and consumption of (high quality) data, and will be organized around three themes:
1) Establishing data needs;
2) Fulfilling data needs with crowd computing; and
3) Evaluating the quality of the retrieved data with respect to the original data need.

Covered topics include:

1) Establishing Data Needs:
- Requirement Elicitation
- Requirement Analysis
- User Modelling Properties

2) Fulfilling Data Needs with Crowd Computation:
- Systems for/with collective intelligence (e.g., recommendation, semiautonomous systems, citizen science, crowdsourcing, and human computation systems)
- Multi-modal Interaction (e.g., conversational systems)
- Human Computation (e.g., worker modelling, task modelling, incentives, task assignment, recruitment)
- Games with a purpose
- Algorithms for Crowd Computing
- Computational Methods for User Modelling
- Interfaces for Crowd Computing Systems

3) Evaluating Retrieved Data:
- Expert Evaluation
- User Evaluation
- Explanation of the output of Crowd Computing Systems

4) Study of Application Domains

When applicable, the course will also feature invited lectures from selected academics and professionals in the field. Since instructors of this course are also directing the Design@Scale Delft AI lab, students of this course will have the opportunity to engage with cutting-edge research projects relevant to this lab.

This Crowd Computing course is an elective for students following the Data Science and Technology Track and the Software Technology Track.
It adds to the master education offer by addressing topics that are complementary to courses like IN4325 Information Retrieval, IN4252 Web Science & Engineering, CS4065 Multimedia Search
and Recommendation, and IN4010 Artificial Intelligence Techniques.

Study Goals

After this course, students will be able to:
- Identify the requirements for a Crowd Computing system [LO1]
- Design and develop Crowd Computing systems. Support and defend the relevance and correctness of his/her choices [LO2]
- Describe and compare several Crowd Computing techniques. [LO3]
- Describe and compare design decisions in the context of Crowd Computing interaction paradigms [LO4]
- Determine which Crowd Computing technique(s) is most appropriate for being used in a certain problem domain [LO5]
- Apply the appropriate Crowd Computing technique to an application domain and evaluate the obtained results. [LO6]
- Analyse the performance of a Crowd Computing system by applying the proper evaluation measures. [LO7]

Education Method

** NB: study guide information may change depending on the developments around the coronavirus.

This course consists of 16 2-hour lectures.
Each week, a 30-minute assignment tests the knowledge acquired on the discussed topics.

Starting from Week 1, students form groups and work on a project, to be presented in week 9. Students are expected to work 6 hours per week (each) on the project assignment.

Expected workload is 32 hours for attending lectures, 24 hours of reading study material and preparing lectures, 55 hours for weekly assignments and group assignment, 24 hours for preparing final survey, and 5 hours for exam and plenary presentations (total 140 hours).

Literature and Study Books:

Materials	- Human Computation. Author(s): Edith Law and Luis von Ahn. Synthesis Lectures on Artificial Intelligence and Machine Learning, June 2011, Vol. 5, No. 3. http://www.morganclaypool.com/doi/abs/10.2200/S00371ED1V01Y201107AIM013 - A. Marcus and A. Parameswaran. Crowdsourced Data Management: Industry and Academic Perspectives. Foundations and TrendsR in Databases, vol. 6, no. 1-2, pp. 1161, 2013. DOI: 10.1561/1900000044. https://people.eecs.berkeley.edu/~adityagp/papers/crowd-book.pdf - An Introduction to Hybrid Human-Machine Information Systems. Demartini, G., Difallah, D.E., Gadiraju, U. and Catasta, M., 2017. Foundations and Trends in Web Science, 7(1), pp.1-87. https://edu.nl/np4th Slides: available on Brightspace Articles: available on Brightspace Recommended reading: - Interaction Design: Beyond Human-Computer Interaction (4th Ed, 2015). Authors: Jenny Preece, Helen Sharp, Yvonne Rogers
Assessment	The final grade consists of the following parts: - Weekly Individual assignment, weighting 15% of the final grade - Group assignment, weighting 55% of the final grade - Final Individual Assignment (Survey), weighting 30% of the final grade The group assignment is performed collectively, but graded individually. Assignments have no re-sit opportunities. Disclaimer: information may change depending on the developments around the coronavirus.
Tags	Algorithmics Artificial intelligence Design Programming Software
Co-Instructor	J. Yang

CS4205	Evolutionary Algorithms	5
Responsible Instructor	Prof.dr. P.A.N. Bosman	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Must have the ability to program your own (extensions of) evolutionary algorithms (e.g., C, C++, Java). Further, although no particular high-level prior knowledge is expected, a basic understanding of key aspects in linear algebra (e.g., matrix multiplications, inversions, decompositions, etc.), probability theory and statistics (e.g., normal distribution, statistical hypothesis testing), algorithm design (e.g., efficient local search, heuristics), complexity theory (e.g., big-O notation, algorithm analysis), will be very helpful.	
Course Contents	In this course we consider a specific subfield of Artificial Intelligence: Evolutionary Algorithms (EAs). These algorithms, sometimes also identified as being part of the class of bio-inspired algorithms, have as a metaphor the concept of natural evolution, i.e., the mechanisms by which, the fittest individuals in a population survive, reproduce, and in doing so, over time, change to be better equipped to thrive in their environment. Initiated in the 60s and 70s of the 20th century, research on EAs has progressed immensely. Today, EAs are being used to solve real-world problems in many areas, e.g. to optimize the layout of electrical wind farms, to automatically create radiation therapy treatment plans, and to optimize the architectures of deep neural networks. This course covers a spectrum of topics in EAs, ranging from basic concepts to advanced, recent, and state-of-the-art research, and ranging from theoretical to applied. In particular, topics include genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, optimal mixing evolutionary algorithms, multi-objective optimization, and real-world applications. The course is planned to have 7 lectures and 2 practical assignments. The first practical assignment pertains to experimenting with already implemented EAs on predefined problems. The second practical assignment offers more freedom, allowing you, in a group, to build your own EA (this may vary depending on student numbers and other circumstances). disclaimer: information may change depending on the developments around the coronavirus.	
Study Goals	Upon successful completion of this course, students will be able to: 1) Explain the key concepts underlying the main streams in Evolutionary Algorithm (EA) research, with in particular genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, and optimal mixing evolutionary algorithms. 2) Explain key ingredients underlying the rationale of when these algorithms work and when they do not work. In particular: schema analysis and how the match between the search bias of an EA and the fitness landscape is influenced by aspects such as variable dependencies and multi-modality. 3) Name and explain key research lines along which state-of-the-art research in EAs is done to achieve more robust, efficient, and effective EAs. 4) Identify good opportunities for using EAs, or hybrid versions thereof, in practice. 5) Properly (scientifically) experiment with EAs as well as program your own.	
Education Method	7 Lectures 2 Lab projects disclaimer: information may change depending on the developments around the coronavirus.	
Literature and Study Materials	Papers and slides that will be made available.	
Assessment	The final grade is based on 60% written exam, 40% lab practical work. disclaimer: information may change depending on the developments around the coronavirus. In particular, there may be no written exam. In that case, there will likely be several smaller practical assignments and a large practical assignment at the end of the course. Both parts will then likely count for 50%. For the large assignment there will also be chance to resit this part by means of a repair option through an oral examination of the lecture contents.	
Permitted Materials during Tests	None	
Tags	Algorithmics Artificial intelligence Optimalisation	

CS4210-B	Intelligent Decision Making Project	5
Responsible Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Dr. J.W. Böhmer	
Instructor	Dr. E. Demirovi	
Instructor	Dr. N. Yorke-Smith	
Instructor	Prof.dr. M.M. de Weerdt	
Contact Hours / Week x/x/x/x	0/0/0/1	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Theoretical knowledge regarding algorithms for decision making in Artificial Intelligence, obtained for instance by passing one of the following courses: - CS4210-A Algorithms for Intelligent Decision Making - CS4400 Deep Reinforcement Learning - IN4010(-12) Artificial Intelligence Techniques - IN4344 Advanced Algorithms.	
Course Contents	Decision making is at the centre of artificial intelligence. Building upon theoretical knowledge gained in other courses, students collaborate in small groups on a distinct research project per group, for instance on decision-making problems in transport, logistics or smart energy grids. Purely algorithmic challenges will also be provided. The research projects provide a good opportunity to learn about topics suitable for Masters projects in the Algorithmics section.	
Study Goals	After completing the Intelligent Decision Making Project course, the student is able to: 1. Apply algorithms for decision making to problem domains, and can compare and evaluate them. 2. Design and implement an extension of a decision-making algorithm. 3. Identify and discuss relevant topics in the research field of algorithms for intelligent decision making. 4. Describe and apply the appropriate research methodology. 5. Communicate his/her findings effectively.	
Education Method	A research project in a small group.	
Literature and Study Materials	Mainly survey papers and book chapters. Details are provided via Brightspace.	
Assessment	The assessment consists of the following items: 1. Quality of work of the research project (40%) 2. A scientific report of the research project (including peer review of a report) (20%) 3. Performance during the project (30%) 4. Oral presentation of the research project (10%) Only items 1 and 2 can be examined a second time. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Only a limited number of students can participate in this course. In order to be admitted, please submit a short motivation letter (max 200 words) via Brightspace. Attending the first lecture is compulsory.	
Tags	Artificial intelligence	
maximum aantal deelnemers	40	

CS4265	Computer and Network Security: Advanced Topics	5
Responsible Instructor	Prof.dr. M. Conti	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	*DISCLAIMER: study guide information may change depending on the developments around the coronavirus.* The course takes the form of seminars based on a selection of scientific papers (that either have had a strong impact on security today, or explore novel ideas that may be important in the future). The list of topics can be found in the brightspace Topics and Papers module. For each topic there is a primary paper, and possibly other additional papers. All the students are required to read all primary papers and be able to competently discuss the material in class. Each student will be responsible for presenting one lecture (based on one of the primary paper including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion in the class. 48 hours before each lecture each student must upload on a shared repository at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). This is intended to be an interactive class: class participation is strongly recommended (and will play a role in the grading criteria). Sleeping during the class is optional, but not recommended.	
Study Goals	This course is about learning to study, analize, do and criticize research in cybersecurity. This will be done by being exposed to actual research topics and scientific papers and discussing things together.	
Education Method	Studying, presenting and discussing recent research results in Computer and Network Security.	
Assessment	Presentation + Class Discussion + Written Report + Oral Exam (please refer to the Judgement field for more details) disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Judgement	The final grade will be made up of four components: 25% the presentation done by the student during the course: each student will be responsible for presenting one topic (based on the corresponding primary paper, including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion (Q&A) in the class. This component is based on following criteria: (15%) Layout and Graphics (30%) Content (20%) Organization (20%) Presentation (15%) Q&A. 25% for the active participation in Q&A sessions during the course: 48 hours before each lecture each student must submit (via email, to both the lecturer and the teaching assistants) at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). The students should actively participate in the discussion of the topics in the 10 minutes Q&A session for each presented topic. 25% for content and quality of the final essay: At the end of the course, each student must write a 5-page long essay about one of the topics that has been discussed in class, or another topic agreed with the lecturer. The topic and the structure of the essay must be agreed with the lecturer. The essay might include some implementation prototype or experiments/simulations to evaluate/support the claim in the paper (in case this is a significant part of the essay, two students can agree with the lecturer to work together). If the student cannot attend the lectures, an alternative work (e.g. a longer essay) must be agreed with the lecturer. 25% for the oral presentation of the essay: during the oral exam, the student is asked to give a 15-minute presentation to the lecturer and the teaching assistants about the essay (presenting with slides is highly recommended). During the oral presentation, students can also be asked questions about other topics of the course. This component is based on following criteria: (30%) Style (20%) Originality (50%) Organization (clarity in your argumentation, coherence between assumptions and conclusions, logical organization, evidence to support claims)	
Co-Instructor	Ir. S.E. Verwer	

CS4280	Language-Based Software Security	5
Responsible Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course has no formal prerequisites. However, for the homework assignments you will have to implement several program analysis techniques using the Scala programming language. If you have not used Scala before, you are thus expected to learn the basics of the language through self-study.	
Course Contents	Security vulnerabilities often arise due to programming errors in the source code of an application. Recent programming errors with severe security implications include Heartbleed (buffer over-read), Shellshock (code injection), and goto-fail (ill-formated code). Rather than hunt for individual vulnerabilities in programs, a more structural approach to improve security is to improve the programming language. This is the goal of language-based security: to rule out whole classes of potential security vulnerabilities in one go. This course studies various security properties and program analysis techniques for enforcing these properties at the level of the programming language to improve software security. In particular, we will study the following properties: - Memory safety: prevent buffer overflows and overreads - Type safety: prevent undefined behaviour - Information flow control: prevent data leaks and code injection attacks We will study techniques to address these problems at the language level through dynamic analysis, static analysis, and language design. To facilitate a precise study and comparison, we will define the above techniques formally in class. To facilitate student experimentation and exploration of trade-offs, students will implement the above techniques in homework assignments.	
Study Goals	After taking this course, students should be able to: 1. Describe the nature and causes of security vulnerabilities in software systems, and give concrete examples of how these security vulnerabilities can be exploited. 2. Explain the properties that can be enforced at the level of the programming language to rule out security vulnerabilities, such as memory safety, type safety, and non-interference. 3. Formally define the semantics of a simple programming language. 4. Formally define dynamic and static analysis techniques for enforcing these security properties. 5. Implement these techniques for a small programming language. 6. Discuss and evaluate the importance of soundness and precision of a given program analysis. 7. Contrast programming languages based on the set of countermeasures they provide, and give an appropriate recommendation for a specific application. 8. Analyse and apply results from scientific literature in the area of language based security.	
Education Method	The course work consists of the following activities: 1 or 2 instruction sessions per week. Weekly homework assignments consisting of theoretical questions, programming assignments, and reading assignments	
Assessment	The assessment for this course consists of two parts: The weekly homework assignments will test your ability to design an implement (variants of) the techniques discussed in the lectures (study goals 3-5). This counts for 40% of the total grade. The final written or oral exam will test your theoretical understanding of the security vulnerabilities and their countermeasures discussed in class (study goals 1-2) and your ability to discuss and contrast the different aspects of these techniques (study goals 6-8). This counts for 60% of the total grade. To pass the course, each of these grades (homework assignments and final exam) should be 5.0 or higher, and the final grade should be 5.8 or higher (and will be rounded to the nearest half grade point). disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	E. Visser	

CS4290	Seminar on Distributed Machine Learning Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	Machine learning systems are often conventionally designed for centralized processing in that they first collect data from distributed sources and then execute algorithms on a single server. Due to the limited scalability of processing large amount of data and the long latency delay, there is a strong demand for a paradigm shift to distributed or decentralized ML systems which execute ML algorithms on multiple and in some cases even geographically dispersed nodes. The aim of this seminar course is to let students learn how to design and build distributed ML systems via paper reading, presentation, discussion, and project prototyping. We provide a broad overview on the design of the state-of-the-art distributed ML systems, with a strong focus on the scalability, resource efficiency, data requirements, and robustness of the solutions. We will present an array of methodologies and techniques that can efficiently scale ML analysis to a large number of distributed nodes against all operation conditions, e.g., system failures and malicious attacks. The specific course topics are listed below. The course materials will be based on a mixture of classic and recently published papers. For each topic, the basic concepts and technology landscape will be first provided and then two state-of-the art of papers will be presented and discussed by students. We offer a testbed of a distributed (deep) ML system in which students are encouraged to apply different techniques to jointly improve its scalability and resilience. Course topics include Overview of distributed machine learning systems Performance and scalability of state-of-the-art systems Acceleration of ML workloads Slim distributed ML systems on small data Robust deep learning systems Federated machine learning systems	
Study Goals	Students are able to argue and reason about distributed ML from a systems perspective. Students understand the behavior and tradeoffs of distributed ML in terms of performance and scalability. Students can estimate the importance of data inputs via different techniques, i.e., core set and decomposition methods, for distributed ML systems. Students understand data poison attacks and design defense strategy for distributed ML systems. Students can analyze the state-of-the art federated machine learning systems and design the failure-resilient communication protocols. Students are able to design and implement methods and techniques for making distributed ML systems more efficient.	
Education Method	Lectures: 7 weeks X 2h Papers: one paper presentation, two paper reviews, and paper discussion. Practical: apply system and algorithmic optimization techniques learned in the lecture to improve the performance of distributed machine learning systems, e.g., image recognition on CIFAR 10. The testbed environment, learning algorithms, and dataset will be given. Deliverables include git commit of functioning code and a report summarizing the contribution	
Assessment	Paper presentation by group (10%): each group of 2 to 3 students needs to choose from a given set a papers to present (15 minutes) and lead the discussion (10 minutes). Paper reviews by individual (30%): each student needs to write three reviews of papers assigned from a given set and those two papers have to be different from the paper for the presented. Each review will account for 10% of the grade. Questionnaires by individual (0%): each student needs to hand in a list of questions at the beginning of the lectures that have paper presentation from other students. This is not graded. Individual project (60%): The project is collaborative among the entire class and competitive as a whole group. The objective is to continuously improve the performance of the given distributed ML system. The students need to hand in a final project report in style of a short scientific paper, stating their individual contribution to the overall system performance. disclaimer: information may change depending on the developments around the coronavirus.	

CS4295	Release Engineering for Machine Learning Applications	5
Responsible Instructor	L. Miranda da Cruz	
Responsible Instructor	Dr.ing. S. Proksch	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	The world of Software Engineering has been revolutionized in the last decade. Instead of releasing software updates yearly, companies can now release multiple times per week, sometimes even per day, to their customers. This allows much quicker reactions to market demands, software failures, and is crucial to increase the business value of software. These improvements have been mostly enabled by advances in release engineering and, in this course, we will learn about the techniques and technologies that build the foundation for modern release engineering. We will go on a journey that starts at continuous integration and then moves on to continuous delivery, continuous deployment, and continuous experimentation. We will discuss the theory and the current research on various related subjects like containerization, testing, or monitoring and will put the learned theory into practice. As a running example, we will build a pipeline for a machine learning application, which -compared to traditional release engineering- poses additional challenges, like data versioning or model deployment.	
Study Goals	After following this course, students are able to... - Apply standard techniques of release engineering - Apply version control techniques to machine learning artifacts, like data or models - Design a deployment pipeline for a machine learning application - Implement quality control techniques in a machine learning pipeline - Analyze and improve existing deployment pipelines - Evaluate and document design decisions in deployment pipelines	
Education Method	- Following interactive lectures - Active participation in tutorial sessions - Reading scientific papers and gray literature - Performing a small literature survey - Implementation of a pre-defined release engineering pipeline - Deriving and implementing an improvement for the pipeline - Documenting the improvement in a scientific essay	
Assessment	Formative Assessment: - Individual group meeting for feedback on current pipeline and pipeline extension proposal - Written feedback on Table of Contents and Introduction of written essay - Individual group meeting for feedback on project progress - Written feedback on methodology and pipeline of written essay Summative Assessment: 35% Final release engineering pipeline (focus: how well is the project executed) 60% Essay (focus: how well have design decisions been evaluated and documented) 5% Presentation (focus: clarification and fraud prevention) Please note: - The different parts of the summative assessment represent grading components and need ALL to be passed to receive a positive overall grade. - There is NO resit opportunity for this course. - Partial grades are not carried over to the next academic year.	
Special Information	The course information presented in the study guide may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. A.E. Zaidman	

CS4410	Category Theory for Programmers	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Categorical structures occur in programming languages on different levels: (1) within programming languages, providing design principles and guidance on how to write modular and correct-by-design programmes (as demonstrated in the practical programming language Haskell) and (2) in the design and study of programming languages, as a guiding meta-theory. In particular, category theory provides a mathematical justification for recursion schemes for inductive datatypes. This course aims to provide solid foundations on both (1) and (2).	
Study Goals	- Use categorical constructions (e.g., monads) in the design and structuring of computer programmes in Haskell - Prove properties of computer programmes, guided by categorical intuition - Understand categorical fusion laws and how to use them to optimize code - Understand the theory of infinite data structures and apply it to practical problems	
Education Method	Learning in this course is achieved through lectures, problem sessions, and guided self-study.	
Assessment	Exam at the end of the term, counts for 100% of the mark.	

EE4715	Array Processing	5
Responsible Instructor	Dr.ir. R.C. Hendriks	
Responsible Instructor	Prof.dr.ir. G.J.T. Leus	
Instructor	Prof.dr.ir. A.J. van der Veen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Linear algebra, signal processing, Fourier transform, stochastic processes and preferably statistical signal processing and some experience with matlab	
Summary	In this course we discuss array processing techniques for signal separation and parameter estimation, using arrays of sensors. After a review/introduction of the necessary linear algebra tools we will start with deriving the signal processing model for narrowband applications, followed by the wideband extension, and apply these to several applications among which array processing for wireless communication, audio and speech processing, biomedical signal processing and astronomy.	
Course Contents	Signal processing models for narrowband and wideband array processing, elementary beamforming concepts (spatial filtering), tools from linear algebra: QR, SVD, eigenvalue decompositions, projections and GEVD. Elementary beamformers/receivers: the matched filter, the Wiener filter, MVDR, LCMV, etc. Estimation of angles and delays using ESPRIT, adaptive space-time filters, the LMS algorithm and factor anaylsis.	
Study Goals	To be able to explain some key problems regarding data models, estimation and detection that occur in array processing applications. - To be able to explain the major signal processing tools required to solve array processing problems. - To be able to implement these signal processing techniques in Matlab. - To be able to apply these techniques to new array processing problems.	
Education Method	Lectures + mini project	
Literature and Study Materials	References from literature and notes	
Assessment	Oral exam: Take-home assignment with oral discussion of the results	

ET4030	Error Correcting Codes	4
Responsible Instructor	Dr.ir. J.H. Weber	
Contact Hours / Week x/x/x/x	0/0/0/3	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	A B.Sc. Programme in Electrical Engineering, Computer Science, or Mathematics	
Course Contents	Introduction into error-correcting codes; mathematical basics; block codes fundamentals; cyclic codes; co-operating codes; soft-decision decoding; convolutional codes; iterative decoding (turbo codes, LDPC codes); applications.	
Study Goals	The global goal of this course is to get acquainted with the basics and applications of error correction coding techniques. Such techniques are applied in order to protect information against errors which may occur during transmission or storage. The specific techniques under consideration in the course are the ones discussed in the lecture notes, which may be updated from year to year according to recent developments. The emphasis will be on the basic trade-offs between efficiency, reliability, and complexity. Unless explicitly indicated, the proofs of the results are not part of the course contents (the interested student may consult books from the bibliography). In the end, the student should be capable of making choices for suitable error correction coding techniques in the context of information transmission and storage applications. The student has to demonstrate to have understood the aforementioned techniques and trade-offs by solving exercises in a closed-book written or oral exam. The level of these exercises is similar to the examples and exercises provided in the lecture notes.	
Education Method	Lectures; expected workload is 22 hours attending lectures, 60 hours preparing for the lectures, studying the lecture notes, and making suggested exercises, and 30 hours for preparing and making the exam.	
Literature and Study Materials	Lecture notes "Error-Correcting Codes" by J.H. Weber	
Assessment	The final grade will be fully determined by a scheduled written exam, which will be held at the end of Q4. If it turns out that this is not possible, then an individual remote oral exam opportunity will be offered instead, also at the end of Q4. The resit will take place as a (remote) oral exam at the end of Q5, on appointment with the lecturer. The exam format is closed-book in any case.	
Remarks	The above-mentioned information may change depending on the developments around the Corona virus. Actual course information available on Brightspace.	

ET4285	Measuring and Simulating the Internet	4
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	(Advanced) Networking course (e.g., CS4055) and Programming skills.	
Course Contents	The Internet is a complex network without a fixed structure. Hence, measuring the Internet is crucial to acquire knowledge about the Internet infrastructure (topology), traffic, and performance (e.g., loss, delay, bandwidth, etc.). This course will discuss the design requirements and challenges in measuring and simulating the Internet, and the existing measurement methodologies (how/where/when to measure). Knowledge of how to conduct and evaluate Internet measurements enables the design and enhancement of a large set of applications, including: capacity planning and traffic engineering, network management and trouble-shooting, detecting network abuse and intrusions, etc.	
Study Goals	The goal of this course is to introduce the students to basic Internet measurement tools, as well as the state-of-the-art in Internet measurements research. The students will learn several Internet measurement techniques (e.g., active vs. passive measurements), and different software tools. Through a measurement assignment, the students will learn how to define/formulate a research problem, choose a specific approach, and complete a measurements-related research project.	
Education Method	Weekly instructions (8x2 hours) + independent project work (8x12 hours).	
Literature and Study Materials	Papers	
Assessment	Groups of students will be assigned a project that requires the students to put the theory on measuring and simulating the Internet into practice. The students have approximately 1 month to complete their assignment. The final assessment is based on the presentation (via report and/or demonstration) of the project assignment results and on the individual contribution and level of participation. Students within a group may thus receive different grades. As this is a project-based course, there is no official resit scheduled. Instead, an opportunity will be given to improve the work.	
Remarks	Disclaimer: The information about ET4285 may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	Because this is a project-based course, we can only admit a limited number of students (typically around 30, but the actual number depends on the number of TAs involved). If more students enrol, we will give preference to those who have successfully completed CS4055.	

IN4185	Globally Distributed Software Engineering	5
Responsible Instructor	Prof.dr.ir. D.M. van Solingen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	Exam by appointment	
Course Language	English	
Expected prior knowledge	Software Engineering (= IN2705)	
Course Contents	The course Globally Distributed Software Engineering (GDSE) will address pro's and con's of GDSE, practical consequences of GDSE, technological (in)feasabilities for GDSE, and practical experiences and examples of GDSE for example in outsourcing, off-shoring, near-shoring and multi-partner systems development. The central theme of this course is the fact that software engineering is carried out in practice more and more in globally distributed settings. This has advantages and disadvantages that need to be addressed in a practical matter when carrying such projects. The course is run asynchronous in BrightSpace. Lectures and excercises are followed digitally in weekly modules that need to be followed prior to the weekly synchronous lecture/virtual meeting. The course hours in the calendar are used for interaction with the professor and more detailed discussion and feedback. The course builds upon individual discipline in preparing the weekly modules online, in combination with group assignments during these weeks as well. Also the group assignments (in groups of 3 or 4 students) can be done virtually.	
Study Goals	The course Globally Distributed Software Engineering (GDSE) aims at teaching participants (1) the technical and organisational setting of carrying out software engineering in practice when distributed over the world, and (2) understanding best-practices in collaboration in software engineering project teams that carry out their work in a distributed setting.	
Education Method	Digital lectures, quizes, group assignments and online discussion. These are used as preparatory work prior to the weekly synchronous lectures (that are merely virtual as well), weekly group home work assignments and individual assignments.	
Computer Use	The course does not contain programming excercies. Though in the group assignment students will have to create a deliverable of choice. This can be very broad from creating a YouTube instruction video to writing an online book, or from creating a Wikipedia page to setting up tooling environment.	
Literature and Study Materials	Presentation handouts	
Assessment	Written report on lab work and literature research, individual f2f examination meeting of 30 minutes with professor. The course grade is calculated from the group assignment (25%), individual essay (25%), personal interview on GDSE course and individual essay (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Please enroll. If enrolled please pay attention that Module 1 of this course needs to be finished PRIOR to the first lecture meeting! Every week a new module is released in BrightSpace that needs to be worked through prior to the weekly synchronous meeting.	
Special Information	Please contact d.m.vansolingen@tudelft.nl	

IN4254	Smart Phone Sensing	5
Responsible Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Requirement 1: Students MUST either (1.1) have passed a JAVA programming course, or (1.2) have passed a C/C++ programming course and be familiar with JAVA, or (1.3) know Objective C (programming language for MACs). This requirement is equivalent to having passed the course TI 1206 in our first year Bachelor curriculum "Object Oriented Programming" Requirement 2: Students MUST (2.1) have passed a basic course on Probability Theory. This requirement is equivalent to having passed the course TI 2216M in our second year Bachelor curriculum "Probability and Statistics". We will be refreshing some concepts on Probability, but we will not be refreshing concepts on Object Oriented Programming.	
Course Contents	The course provides an introduction to the current research trends in the area of smartphones. The course will be based on a programming project, where students will form groups of two to develop a smartphone application. This is not a programming course; students are expected to have already programming experience. To develop a smartphone application, a user needs to be familiar with (1) the signals and data that smartphones can gather, and (2) the mathematical tools necessary to process this data. This course will provide a solid background for the above two points. During the lectures we will analyze the latest research papers on this emerging field. We will dissect these papers to understand how techniques from algorithms, signal processing and machine learning are used to develop some exciting applications. The students will then use these basic technical tools to develop their own apps.	
Study Goals	The goals of this course are twofold. First, to expose students to the increasingly important area of mobile computing. Students will learn how mobile phones can be used to solve problems in areas ranging from health care and indoor localization to song recognition and traffic management. Second, to provide students with a basic set of tools to develop their own applications. For students aiming for industry, the course should enhance their ability to use theoretical tools to solve practical problems. For students involved on research activities, the course will provide them with the necessary background to use smartphones as a distributed sensing and processing unit that could be used to solve the particular problems in their areas. After taking this course students will be able to: (1) Explain the current applications, methods and research trends in the area of smartphone sensing. (2) Apply key mathematical tools in the development of smartphone applications. (3) Analyze how a sensing and computing problem can be solved via the use of smartphones, and identify the steps required to design a solution. (4) Create a non--trivial and innovative smartphone application.	
Education Method	Lectures + Lab The project work, including the written report, covers the entire duration of the course period, and will take approximately 120 hours, of which 14 hours are spent on lectures, 10 hours preparing reports, 10 hours reading research papers, and the remaining part programming the App (the time spent in the Lab belong to this latter part).	
Literature and Study Materials	Research Papers and web tutorials	
Assessment	Written reports + project presentation + oral exam Overall, the final grade is determined by: 1) Two intermediate reports (5% of grade each, 2 pages each) 2) Final report (10 % of grade, 5 pages) 3) Final project demonstration (80% of grade) The first two reports are due on the third and fifth week; and the final report, project and exam are due on the ninth week. There is no resit for this course. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	1. You need to enrol in Brightspace 2. The first lecture will be compulsory 3. This course can only accommodate 60 students, with ES students having a preference when demand exceeds capacity. If your program marks this course as required, you are guaranteed a spot. IMPORTANT: The study guide information may change depending on the developments around the coronavirus.	

IN4255	Geometric Data Processing	5
Responsible Instructor	Dr. K.A. Hildebrandt	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Basic knowledge in mathematics (linear algebra, calculus): TI1106M, TI1206M or comparable courses. Students who haven't followed any of these courses can follow the course, but should be willing to invest more time.	
Course Contents	Geometry processing is concerned with the representation, analysis, manipulation, and optimization of digital shapes. Thanks to the advances in 3D acquisition and manufacturing technologies (like 3D-Scanning and 3D-printing), the usage of geometric data is continuously increasing and an efficient processing of digital shapes plays an important role for a variety of applications in areas such as computer graphics, computer-aided design and engineering, medical imaging and surgery planning, architecture, and entertainment. In this course, we will study concepts and algorithms for creating, analyzing, editing and optimizing digital geometric shapes.	
Study Goals	After successfully completing this course, the student is able to: - describe the fundamental techniques used for representing, analyzing, processing and modeling digital 3D-shapes treated in the course and to explain the mathematical and algorithmic concepts associated with them - apply the learned mathematical concepts to solve basic geometric problems arising in geometric modeling applications - design algorithms that can solve simple geometric modeling tasks and evaluate the drawbacks, benefits and limitations of the proposed algorithms - implement the designed algorithms in a geometric modeling software framework	
Education Method	The course combines lectures, tutorials, practical project work, and homework assignments.	
Literature and Study Materials	References to textbooks and recent research and survey papers are given in the lectures.	
Assessment	The course will be assessed on two practical projects and two theoretical assignments. The course grade is a weighted average of the grades of the practical projects (60%) and the theoretical assignments (40%). Note that, there is a minimum grade of 5.0 for each assignment grade and the average grade for all components of at least a 5.8 in order to pass the course. Also, grades for individual assignments do not carry to the next year. Resubmission of modified coursework is only allowed for assignments that received a fail grade (<5.0). Overall resit grades will be capped to 6.0 disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4331	Web-scale Data Management	5
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Dr. A. Katsifodimos	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Bachelor level courses in database management systems and operating systems. A prior course in distributed systems or middleware would be helpful but is not required. Programming skills are important for the final assignment.	
Course Contents	This course addresses the challenges of Data Management at Web-scale. Especially, it covers the need for large-scale distributed data storage systems. The lecture therefore introduces step-by-step increasingly complex distributed storage systems, leading up to modern implementations of different NoSQL data storage systems. The challenges arising from such systems are presented and discussed, especially focusing on the CAP theorem and the resulting trade-offs with respect to data models, transactional power, query expressivity, and replication consistency. These discussions lead to different variants of NoSQL database systems, like Key-Value Stores, Document Stores, Wide-Columnar stores, and Graph Databases. The advantages, disadvantages, and general properties of these systems are discusses in more detail. There is special focus on distributed transactions and consistency guarantees of different data management systems and methods.	
Study Goals	At the end of this course the student can - assess the nature of a given storage problem, and can select a suitable technology for solving it - understands the different data models encountered in Web Data Management, and their impact on modelling and querying - understands the issues arising from distributing and replicating data, especially with respect to the CAP theorem -understands the trade-offs which can be chosen within the design space of the CAP theorem -categorize and explain modern NoSQL databases within the framework of the previously mentioned trade-offs	
Education Method	Lectures and assignments	
Literature and Study Materials	Course slides and Lecture Videos	
Books	Literature mentioned in the lecture, mostly research papers.	
Assessment	Depending on the year, there are group assignments with final presentation, written report, and oral interviews. Final grade will be the weighted average of the three partial grades (the weights will be announced during the lecture depending on the available topics and assignments in the current quarter). The interview covers questions from the topics of the lecture. The interview can be resit, the assignment & presentation cannot. disclaimer: information may change depending on the developments around the coronavirus.	

IN4333	Language Engineering Project	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/0/0/4 (lab)	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Compiler construction CS4200-A and CS4200-B.	
Course Contents	"Software systems are the engines of modern information society. Our ability to cope with the increasing complexity of software systems is limited by the programming languages we use to build them. Bridging the gap between domain concepts and the implementation of these concepts in a programming language is one of the core challenges of software engineering. Modern programming languages have considerably reduced this gap, but often still require low-level programmatic encodings of domain concepts. Or as Alan Perlis formulated it in one of his famous epigrams: 'A programming language is low level when its programs require attention to the irrelevant'. A fixed set of (Turing Complete) programming constructs is sufficient to express all possible computations, but at the expense of considerable encoding that obfuscates the concepts under consideration. Linguistic abstraction can be used as a tool to capture our emerging understanding of domains of computation." (Visser, SCP 2015) In the precursor compiler construction course (CS4200), students learn the basics of language engineering by building a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts. In the precursor compiler construction course (IN4303), students learn the basics of language engineering and build a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts.	
Study Goals	In this course students learn to apply language engineering principles and tools to a real (domain-specific) programming language. Explore the definition of all aspects of a programming language: syntax, name binding, type analysis, transformations, code generation.	
Education Method	This is a project course. Students deepen their language engineering skills and insights by building a complete language definition. Students work in teams of two on the definition of a (domain-specific) programming language using the Spoofax Language Workbench. Assistance and feedback is provided during weekly lab hours. The project should span the full life cycle of language implementation including a test suite, IDE, code generator, and distribution of the result as an Eclipse plugin.	
Literature and Study Materials	- Documentation of the design and implementation of a specific language - Papers about language definition techniques	
Assessment	The work is assessed based on a code review of the language definition, a written report about the project, and a presentation in the final project workshop. The course has no resit. disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is based on the following components: - git repository with language project (40%) - written report about the project (30%) - presentation (slides) (30%)	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Suggested Track Courses AIT 2021

IN4089	Data Visualization	5
Responsible Instructor	T. Höllt	
Instructor	Prof.dr. E. Eisemann	
Contact Hours / Week x/x/x/x	0/2/0/0 & lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	Master course MKE/ST/DS	
Expected prior knowledge	IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required). The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.	
Course Contents	Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization, The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines. As a computer science course, affinity to algorithmic thinking and programing skills will be needed. Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.	
Study Goals	The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data. Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more. By the end of the course, you should be able to LO1: Discuss a large range of visualization techniques. LO2: Discuss a perception principle of visualization. LO3: Explain mathematical principles and algorithms of visualization techniques. LO4: Design suitable visualization systems for a given practical data analysis problem. LO5: Implement visualization systems for a given practical data analysis problem.	
Education Method	Lectures, practical assignments, self-study, and projects.	
Literature and Study Materials	Course slides, instructions for projects, and selected literature. Chapters from: Visualization Analysis and Design Author: Tamara Munzner CRC Press Visual Computing for Medicine 2nd Edition Theory, Algorithms, and Applications Authors: Bernhard Preim Charl Botha Morgan Kaufmann All available in electronic form via Brightspace or at TUDelft library.	
Assessment	The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Judgement	The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam. The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project. Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions. The project is evaluated based on the developed result, its documentation and presentation. Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Seminar Courses CS & Literature Survey 2021

CS4120	Seminar Science and Methods in Cyber security	5
Responsible Instructor	Dr. S. Picek	
Instructor	Dr. M.P.M. Franssen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	This seminar course Cyber Security covers the following topics: (i) an introduction to the philosophy of (classical and design) science, (ii) the art of writing a scientific research proposal, (iii) an overview of useful and relevant scientific methods, (iv) introduction to scientific writing (of a paper and of a MSc thesis).	
Study Goals	1. Getting a basic knowledge and understanding of what science entails and how scientific knowledge is being created 2. Getting knowledge and understanding of relevant scientific methods applicable in the field of Cyber Security 3. Getting knowledge, understanding and skills for writing a research proposal related to the creation of a MSc thesis 4. Getting knowledge and understanding on how to execute a scientific article and MSc thesis 5. Getting knowledge and understanding of how to execute a literature review.	
Education Method	Lecturers supported by the execution of mostly individual assignments. Attendance of participants in this course is mandatory.	
Assessment	Final grade will be based on a weighted average of various scores including (i) presence and level of participation (10%), (ii) quality of the research proposal to be written and presented (60%), (iii) grades for assignments (paper evaluation, paper rewrite, essay questions/written exam) (30%). No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Research Methods	

CS4125	Seminar Research Methodology for Data Science	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Instructor	Dr. K.A. Hildebrandt	
Instructor	J. Urbano Merino	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	basic knowledge in mathematics (linear algebra, calculus, probability and statistics)	
Course Contents	The course focuses on research methods for data science. It looks at underlying principles and concepts for data collection, analysis and data processing, as well as the use of tools to do this. The main topics of study are: Conceptualizing research questions and experimental design Frequentist and Bayesian data analysis Generalized linear models for statistical analysis Multilevel modelling for hierarchical and longitudinal data analysis Measuring and sampling, validity and reliability Linear and nonlinear dimensional reduction Principles of statistical testing In the course, students will be using software tools such as R, and Matlab/Mathematica	
Study Goals	The main aims of this module for the student is to achieve understanding of research methods for data science and obtain practical experience with data analysis and data processing methods. This module provides students with the opportunity to develop and demonstrate their understanding, knowledge, and competence. The learning outcomes for the module are that students will be able to: 1. Appreciate and comprehend strategies for collecting and processing data to answer data-driven research questions 2. Understand and reproduce key principles underlying statistical data and data processing analysis 3. Learn to identify and avoid typical biases, paradoxes and misunderstandings in data-driven research 4. Apply and select appropriate data modelling techniques to analyse data and data processing	
Education Method	Lectures/Assignments Expected Workload Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours (5 × 5 hours for each tool) Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours) Total = 140 hours	
Literature and Study Materials	Will be provided online	
Assessment	Course will be assessed on 3 coursework assignments. A) Analysis of experimental research data (40%) B) Exploration of real-world data set (20%) C) Linear and nonlinear dimensional reduction (40%) Students work in small groups on the 3 assignments. For each assignment, the student group submit a report and give a presentation including a question and answer round where individual group members are assessed on the coursework. The final course mark is the weighted average of the three assignment marks. Note that, there is a minimum grade of 5.0 for each assignment grade and an average grade for all components of at least a 5.8 in order to pass the course. Also, marks for individual assignments do not carry to the next year. Resit next quarter Resubmission of modified coursework is only allowed for assignments that received a fail mark (<5.0). Overall resit mark will be capped to 6.0. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	NA	

CS4130	Seminar Programming Languages	5
Responsible Instructor	E. Visser	
Responsible Instructor	Dr. S.S. Chakraborty	
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Dr. J.G.H. Cockx	
Instructor	C.B. Poulsen	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Followed at least one other Programming Languages master course	
Course Contents	Programming languages is a core field in computer science that studies the design, theory and applications of both new and existing programming languages. Topics in programming languages include compiler construction, program analysis, program transformations, meta programming, parsing, formal semantics, program verification, and type systems. In this course, we will read scientific journal and conference articles in the field of programming languages to get a deeper understanding of programming languages. If you wish to do a MSc thesis in the programming languages group, we highly recommend taking this course.	
Study Goals	The student will acquire: - Skills to read and discuss scientific articles. - Understanding of the topics in the research field of programming languages. - Understanding of the research methodology in the research field programming languages.	
Education Method	We will run this seminar as a discussion seminar with meetings twice a week. In each meeting, we discuss a scientific article that has been studied by the participants in advance. The following activities are required for each meeting: - Reading a scientific article - Writing and submitting a short summary of the article (max 0.5 pages) - Active participation in the discussion of the article Expected Workload: - 4h Discussion sessions - 6h Reading paper at home - 2h Writing summary at home	
Literature and Study Materials	Papers from the programming languages literature will be assigned at the start of the course	
Books	No books	
Assessment	Students get a grade for each meeting based on the participation in the discussion. The final grade is the average of the grades for the meetings. There will not be a resit for the course. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	not applicable	
Enrolment / Application	The number of participants for this course is limited. Students in the second year of the master and students that follow the Programming Languages specialization have priority.	
Judgement	The final grade is the average of the grades for the meetings.	

CS4165	Seminar Social Signal Processing	5
Responsible Instructor	Dr. H.S. Hung	
Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Contact Hours / Week x/x/x/x	2/2/0/0 + project	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Your background should consist of a combination of at least two of these topics or related topics: Signal Processing, Speech/Audio Processing, Computer Vision, AI, Machine Learning, Pattern Recognition, Reinforcement Learning, Deep Learning/ Neural Networks, Cognitive Modelling. These can be topics that you learned about at either Bachelor or Master level.	
Course Contents	The core of social intelligence is our ability to understand and interpret social signals of a person we are communicating with is. Social intelligence is a facet of human intelligence that has been argued to be indispensable and perhaps the most important for success in life. Social Signal Processing (SSP), the new, emerging, domain aimed at understanding social interactions through machine analysis and production of nonverbal behavior. In this course you will learn how next-generation computing can make use of such social signals by giving it the ability to recognize and produce human social signals and social behaviors. Think about turn taking, politeness, disagreement, emotions, rapport. You will learn about relevant findings in social psychology, and you will learn computational techniques that allow systems to make use of social signals to become more effective and more efficient by being able to detect but also simulate (e.g. in virtual agents) blinks, smiles, crossed arms, laughter. Socially aware computing. These techniques can be used in robots, virtual agents, smart homes, crowd monitoring, etc.	
Study Goals	Know what social signals are. Be able to apply computational methods to detect and simulate such signals. Position the field of social signal processing in computer science and psychology, and identify its major goals and angles of study. Define and explain social signals in humans and know about major psychological theories of social interaction. Explain major social signal recognition, simulation and expression techniques in computational systems. Develop (in groups) a research project that uses social signals in a non-trivial manner with hypotheses, research questions, and a supporting literature survey, and together evaluate the resulting system. These are important skills that prepare students towards their own masters thesis study later on.	
Education Method	This course is run as is a two quarter course running in Q1 and Q2. The course has been historically open to students from Leiden University as we see that mixed university groups leads to better quality projects and peer learning, which is a key part of the course. The course has therefore been designed to block off Monday afternoons where lecture times are scheduled to allow travel between the two institutions as part of the timetabled course. In light of the move to have campus education again, the lectures will be on campus and not virtual. The course has historically been run also as a Q1 only course. However, we have found that the 2 quarter model allows students more time to learn and absorb the course learning objectives leading to higher quality projects. In practice most of the contact hours are in Q1 with more time being devoted to the project work in Q2 with occasional progress meetings. Seminar: 2 hours of lectures per week for most of Q1. Self-study of papers. The papers will be made available at the start of each lecture. Project: 2 hours of class contact hours every 2-3 weeks for latter part of Q1 and Q2. Perform a piece of research (survey, research question, programming, testing) and write a paper about it. Students will work in teams of about 3-4 persons! Depending on the total number of students enrolled, teams will either work on a topic of their own or we will all together work on one big topic.	
Literature and Study Materials	Selected papers made available before the course.	
Assessment	UPDATED as a result of on campus education. 10% mini exam. This multiple choice assessment helps to establish basic knowledge of the course material before the project work starts. 40% project proposal quality (mid term presentation + survey) 50% Project execution. No final exam. disclaimer: information may change depending on the developments around the coronavirus.	
Maximum number of participants	60	

CS4210-B	Intelligent Decision Making Project	5
Responsible Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Dr. J.W. Böhmer	
Instructor	Dr. E. Demirovi	
Instructor	Dr. N. Yorke-Smith	
Instructor	Prof.dr. M.M. de Weerdt	
Contact Hours / Week x/x/x/x	0/0/0/1	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Theoretical knowledge regarding algorithms for decision making in Artificial Intelligence, obtained for instance by passing one of the following courses: - CS4210-A Algorithms for Intelligent Decision Making - CS4400 Deep Reinforcement Learning - IN4010(-12) Artificial Intelligence Techniques - IN4344 Advanced Algorithms.	
Course Contents	Decision making is at the centre of artificial intelligence. Building upon theoretical knowledge gained in other courses, students collaborate in small groups on a distinct research project per group, for instance on decision-making problems in transport, logistics or smart energy grids. Purely algorithmic challenges will also be provided. The research projects provide a good opportunity to learn about topics suitable for Masters projects in the Algorithmics section.	
Study Goals	After completing the Intelligent Decision Making Project course, the student is able to: 1. Apply algorithms for decision making to problem domains, and can compare and evaluate them. 2. Design and implement an extension of a decision-making algorithm. 3. Identify and discuss relevant topics in the research field of algorithms for intelligent decision making. 4. Describe and apply the appropriate research methodology. 5. Communicate his/her findings effectively.	
Education Method	A research project in a small group.	
Literature and Study Materials	Mainly survey papers and book chapters. Details are provided via Brightspace.	
Assessment	The assessment consists of the following items: 1. Quality of work of the research project (40%) 2. A scientific report of the research project (including peer review of a report) (20%) 3. Performance during the project (30%) 4. Oral presentation of the research project (10%) Only items 1 and 2 can be examined a second time. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Only a limited number of students can participate in this course. In order to be admitted, please submit a short motivation letter (max 200 words) via Brightspace. Attending the first lecture is compulsory.	
Tags	Artificial intelligence	
maximum aantal deelnemers	40	

CS4245	Seminar Computer Vision by Deep Learning	5

Responsible Instructor: Dr. J.C. van Gemert

Instructor: S. Pintea

Contact Hours / Week x/x/x/x: 0/0/0/8

Education Period: 4

Start Education: 4

Exam Period: 4
5

Course Language: English

Required for: MSc thesis in the Computer Vision lab

Expected prior knowledge: Deep Learning (CS4240)

Course Contents

The recent boom in computer vision and automatic image understanding represents an inflection point in human productivity, permeating wide aspects of the economy and society. Examples of visual tasks which are repetitive or require expert knowledge include medical diagnosis, industrial inspection, autonomous vehicles, etc. When machines can meaningfully assist or even completely take-over such tasks it will change the world as we know it.

The breakthrough in the 2012 ImageNet automatic image recognition competition shows all previously existing methods decisively defeated by a deep neural network. Deep learning replaces feature engineering methods and is able to successfully learn image features from huge annotated datasets.

This course is on automatically understanding visual content such as images and videos by deep learning.

Topics include: Fundamentals in Vision, object detection, per-pixel labelings, video recognition, image similarity learning, efficiency, self-supervision, 3D computer vision, adverserial attacks, explainability, generative models.

The course will have lectures, a seminar and a lab practical:

- The lectures will be on established topics based on the current literature.

- The seminar will have students read, critique, and present relevant computer vision research papers. You will have to read 2 papers per week, for 7 weeks (14 papers).

- The lab will have students apply and design their own (small) computer vision project.

The course build on top of the Deep Learning course (CS4240) and follows a similar setup.

Study Goals

Upon successful completion of the course, students will be able to:

[LO1]. Describe the deep learning techniques reviewed in the course for computer vision applications such as image classification, object detection, per-pixel labelings, video recognition, image similarity learning.

[LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers

[LO3]. Debate upon positive and negative aspects of techniques and research papers

[L04]. Quickly identify the core contributions of a research paper

[LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (we focus on Pytorch)

[LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain.

[LO7]. Apply the appropriate technique to a (simple) Computer Vision problem.

[LO8]. Write clearly and concisely about your code, method, results, and analyis.

Education Method

Lectures

Lab project: design and execute your own Computer Vision project.

Seminar: paper reading, critiquing, and presenting.

Assessment

1. Presentation: during the seminar a small group of students presents a paper. You will have to present once.

2. Students will have to submit relevant questions about papers/lectures

3. Lab assignment: in a small group of students you work on a deep learning project.

4. Exam about the papers and the theory.

*** Disclaimer: Assessment this year may change depending on the COVID19 virus ***

CS4285	Seminar: Decentralized Systems	5
Responsible Instructor	Dr. S. Roos	
Instructor	Prof.dr. J.S. Rellermeyer	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	Systems with one central party enable large-scale surveillance, suffer from a lack of reliability, and open the door to censorship and easy manipulation of public opinion with severe consequences for, e.g., elections. Decentralization avoids or at least mitigate these problems but comes with a number of challenges such as maintaining high performance and legal compliance. In this course, we first discuss advantages and disadvantages of decentralization. Afterwards, we group decentralized systems into three levels: 1) systems that add a decentralized component to centralized systems to enhance privacy (e.g., anonymity networks); 2) systems that have no central servers but still fully depend on the standard Internet architecture to work (e.g., blockchains), 3) systems that have no central servers and do not (only) use the Internet architecture (e.g., ad-hoc networking). Students will gain an overview of approaches and concrete systems in all three categories. They will further evaluate the introduced systems with regard to security, privacy, and performance. Course topics include Consequences and challenges of decentralization Onion routing, its implementation in Tor, and challenges faced by Tor Techniques to resist censorship and their impact on performance Methods for structuring overlay networks Anonymous and censorship-resistant overlay networks such as Freenet Censorship-resistant blockchain networks Methods for communicating without (directly) connecting to the Internet	
Study Goals	Define key concepts in the field of decentralization Describe the positive and negative impacts of decentralization on security, privacy, and performance of applications Explain and assess the key algorithms of deployed decentralized systems Apply mathematical proofs, simulations, or real-world measurements to evaluate decentralized systems	
Education Method	1. Lectures: 7 weeks X 2h 2. Paper reviews: Students read papers and come up with a survey. The course is blended and knowledge of the content of the papers is necessary to follow the subsequent lecture in-depth. 3. Practical: Students have two homework assignments where they evaluate a given decentralized system with regard to its privacy, security, or performance. 4. Presentation: The course contains a presentation of the conducted work	
Assessment	1. Paper reviews/survey (50%): The student will survey a set of papers 2. Practicals (40%): There will be two homework assignments, each worth 20%, to be submitted in week 4 and 8. Students have three weeks to complete the homework. 3. Presentation (10%): The student presents their work during the last week. There will be no classical exam and no resit for the practical assignments. Partial grades do not carry to the next year.	

CS4290	Seminar on Distributed Machine Learning Systems	5

Responsible Instructor	Dr. Y. Chen
Contact Hours / Week x/x/x/x	0/0/0/2
Education Period	4
Start Education	4
Exam Period	none
Course Language	English
Course Contents	Machine learning systems are often conventionally designed for centralized processing in that they first collect data from distributed sources and then execute algorithms on a single server. Due to the limited scalability of processing large amount of data and the long latency delay, there is a strong demand for a paradigm shift to distributed or decentralized ML systems which execute ML algorithms on multiple and in some cases even geographically dispersed nodes. The aim of this seminar course is to let students learn how to design and build distributed ML systems via paper reading, presentation, discussion, and project prototyping. We provide a broad overview on the design of the state-of-the-art distributed ML systems, with a strong focus on the scalability, resource efficiency, data requirements, and robustness of the solutions. We will present an array of methodologies and techniques that can efficiently scale ML analysis to a large number of distributed nodes against all operation conditions, e.g., system failures and malicious attacks. The specific course topics are listed below. The course materials will be based on a mixture of classic and recently published papers. For each topic, the basic concepts and technology landscape will be first provided and then two state-of-the art of papers will be presented and discussed by students. We offer a testbed of a distributed (deep) ML system in which students are encouraged to apply different techniques to jointly improve its scalability and resilience. Course topics include Overview of distributed machine learning systems Performance and scalability of state-of-the-art systems Acceleration of ML workloads Slim distributed ML systems on small data Robust deep learning systems Federated machine learning systems
Study Goals	Students are able to argue and reason about distributed ML from a systems perspective. Students understand the behavior and tradeoffs of distributed ML in terms of performance and scalability. Students can estimate the importance of data inputs via different techniques, i.e., core set and decomposition methods, for distributed ML systems. Students understand data poison attacks and design defense strategy for distributed ML systems. Students can analyze the state-of-the art federated machine learning systems and design the failure-resilient communication protocols. Students are able to design and implement methods and techniques for making distributed ML systems more efficient.
Education Method	Lectures: 7 weeks X 2h Papers: one paper presentation, two paper reviews, and paper discussion. Practical: apply system and algorithmic optimization techniques learned in the lecture to improve the performance of distributed machine learning systems, e.g., image recognition on CIFAR 10. The testbed environment, learning algorithms, and dataset will be given. Deliverables include git commit of functioning code and a report summarizing the contribution
Assessment	Paper presentation by group (10%): each group of 2 to 3 students needs to choose from a given set a papers to present (15 minutes) and lead the discussion (10 minutes). Paper reviews by individual (30%): each student needs to write three reviews of papers assigned from a given set and those two papers have to be different from the paper for the presented. Each review will account for 10% of the grade. Questionnaires by individual (0%): each student needs to hand in a list of questions at the beginning of the lectures that have paper presentation from other students. This is not graded. Individual project (60%): The project is collaborative among the entire class and competitive as a whole group. The objective is to continuously improve the performance of the given distributed ML system. The students need to hand in a final project report in style of a short scientific paper, stating their individual contribution to the overall system performance. disclaimer: information may change depending on the developments around the coronavirus.

IN4306	Literature Survey	10
Responsible Instructor	Dr.ir. W.P. Brinkman	
Responsible Instructor	L. Miranda da Cruz	
Contact Hours / Week x/x/x/x	Not applicable	
Education Period	None (Self Study)	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	The Literature Survey is an individual assignment carried out under the supervision of a Computer Science staff member, i.e. an assistant, associate or full professor. For this assignment the student reads a broad range of papers in the chosen specialisation field and writes a report in which the ideas found in the papers are discussed and compared. It is not allowed to merge this assignment with the thesis project.	
Study Goals	The student is able to read contemporary scientific literature in the chosen field of specialisation. The student is able to distill the main ideas of a paper and to write these down in his or her own words. The student is able to place the ideas of different papers in perspective by comparing these. The student is aware of the most important academic journals and conferences of the research field of the chosen specialization. The student understands the role of communication and writing inherent in academic research (e.g. peer review process). The student understands experimental principles (hypothesis, validation, evaluation, theoretical versus empirical results, .).	
Education Method	Individual assignment and individual guidance by a scientific staff member.	
Assessment	Writing a scientific report, individually and under supervision of a staff member. This staff member will also mark the report. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	The Literature Survey may be part of an individual exam programme of a student, which has to be approved by the Board of Examiners (BoE). To apply for a literature study the student should contact a staff member of the research group of her/his chosen specialisation after having received approval of his or her individual exam programme. The staff member and the student make arrangements regarding content and scope of the survey. The abovementioned staff member will supervise the student during this course.	
Co-Instructor	Dr.ir. A.R. Bidarra	
Co-Instructor	Dr. T.E.P.M.F. Abeel	
Co-Instructor	Dr. K.A. Hildebrandt	
Co-Instructor	Dr. J.C. van Gemert	
Co-Instructor	Dr. C. Lofi	

IN4310	Seminar Computer Graphics	5
Responsible Instructor	Dr.ir. A.R. Bidarra	
Instructor	Dr. R. Marroquim	
Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. K.A. Hildebrandt	
Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	One of the CS core courses (IN4086 Data Visualization, and IN4152 3D Computer Graphics and Animation), and at least one of the Computer Graphics specialization courses (IN4255 Geometric Modeling, IN4302 Building Serious Games, and IN4307 Medical Visualization) are expected as prior knowledge.	
Course Contents	In this seminar you work on a selection of recent topics and results in one of the areas of Computer Graphics.	
Study Goals	To obtain in-depth knowledge about an advanced topic within Computer Graphics, in particular in rendering, game technology, visualization, appearance capture, animation or geometric modeling. The seminar may be used as a preparation for an MSc thesis topic. The student will acquire practical skills in reading, presenting, explaining, and discussing scientific papers, as well as writing scientific papers.	
Education Method	This course has the format of a student seminar. Students will prepare a scientific presentation of a recent research paper. The presentation goes in-depth and covers the papers strengths and weaknesses. After each presentation, a research discussions takes place and will be held in the plenary colloquium sessions. Students will participate in a scientific discussion of some of the presented papers. Finally, each student will realize a particular aspect of the chosen paper in form of an implementation, which will be presented in a dedicated session and described in a short document (1 page + figures) following the structure of a scientific article.	
Literature and Study Materials	Recent research papers about the selected topic.	
Assessment	The course will be assessed based on several components: - Presentation of a recent scientific paper (50%) - Implementation component reproducing an aspect of the paper (40%, including short report and presentation) - Preparation of questions for some of the presented research papers (10%) The course is passed if all components have at least a grade of 5.0 and the weighted average is a 6 (rounded on 0.5). disclaimer: information may change depending on the developments around the coronavirus.	

IN4314	Seminar Selected Topics in Multimedia Computing	5
Responsible Instructor	P.S. Cesar Garcia	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	signal (image, audio) processing, pattern recognition, networking and distributed systems	
Course Contents	Through all the exciting recent advances in digital media technology and the rapid growth of social media platforms, multimedia content is increasingly embedded in our daily lives, gaining enormous potential in improving the traditional educational, professional, business, communication and entertainment processes. To be able to use this potential for transferring these processes into user-centric interactive multimedia applications, technology is required that can help us access, deliver, enrich and share rich-media content. This course provides insight into the state-of-the-art cross-disciplinary research efforts related to the development of such technology. The topics covered by the course include, but are not limited to, multimedia systems (transport and delivery, telepresence and VR, mobile), multimedia experiences (Quality of Experience, Collaboration), and multimedia engagement (emotional and social signals and social multimedia).	
Study Goals	To become acquainted with the state-of-the-art research and development activities in the field of Multimedia Computing, and to become an expert in one particular "hot topic", such that they are able to identify the "knowledge gap" (i.e., the place in which more research is needed in order to advance the state of the art).	
Education Method	readings, seminar discussions, presentations, survey paper	
Literature and Study Materials	Readings, possibly including video lectures.	
Assessment	The students demonstrate the knowledge that they have acquired by making a presentation on a pre-existing survey (10%), then writing their own survey on a new topic (65%), and finally by making a presentation on that topic (25%). The students must complete all three components. disclaimer: information may change depending on the developments around the coronavirus.	

IN4326	Seminar Web Information Systems	5
Responsible Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Standard bachelor-level computer science or equivalent.	
Course Contents	In this course we discuss recent developments in the area of web information systems. We select topics to discuss from the areas of: - web technology (e.g. web engineering, hypertext, adaptive web), - web data management (e.g. web data interoperability, system and data integration), - web data and semantics (e.g. ontologies, semantic web, metadata), - web data analytics (e.g. user modeling, web personalization, web information filtering and retrieval), - social web (e.g. social web data analytics, social networking, human computing), - web science (e.g. crowdsourcing, trust, data science). We discuss this content while learning about the role of scientific communication and about the scientific methodologies and approaches for conducting research in the area. In this seminar the students will have to prepare and give scientific presentations on the basis of research papers about selected topics - the topics are selected in the first session together with the students. The students will also have to attend the presentations and participate in discussions on the presented papers. In addition, students will have to write a short survey about a topic in the area of web information systems of their choice.	
Study Goals	- to expose the student to current developments in research on web information systems and be aware of the methodologies and approaches to conduct research in the area; - to familiarise the student with reading, presenting and discussing scientific literature in the area and be aware of the most important academic journals and conferences in the area (and their review processes); - to help the student in reading and writing scientific papers and choosing a topic for her/his thesis in the area.	
Education Method	Student seminar.	
Literature and Study Materials	Is provided in the seminar, depending on the chosen subjects.	
Assessment	- Quality of presentation of the scientific paper studied (15%). - Participation in the seminar discussions (10%). - Quality of paper written (75%). disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace beforehand. Students are also asked to be present and active in the first seminar session, to facilitate the proper planning of the seminar.	
Remarks	The expected workload of 5ects is distributed uniformly over the quarter. The seminar asks for active participation and therefore can only be completed as part of the first quarter edition; there is no re-sit.	
Tags	Artificial intelligence Databases	
Maximum number of participants	This course has a maximum capacity of 50 students. Students of 1) Web Information Systems group and 2) EEMCS have priority for other students.	

IN4334	Analytics and Machine Learning for Software Engineering	5
Responsible Instructor	M. Finavaro Aniche	
Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	- Experience with programming is required - Experience with research methods is nice to have - Experience with statistics / machine / deep learning is nice to have	
Course Contents	Software repositories archive valuable software engineering data, such as source code, execution traces, historical code changes, mailing lists, and bug reports. This data contains a wealth of information about a projects status and history. Doing data science on software repositories, researchers can gain an empirically-based understanding of software development practices, and practitioners can better manage, maintain, and evolve complex software projects. Moreover, the advances in Machine Learning and AI technologies, as demonstrated by the successful application of Deep Neural Networks in various domains did not go unnoticed in the field of Software Engineering; researchers have applied machine learning to tackle different software engineering tasks. In this seminar, we will explore different software analytics tools and techniques to investigate different software engineering phenomena, and machine learning / deep learning models that tackle software engineering problems.	
Study Goals	IN4334 is a seminar course that aims to give students a deep understanding of and hands-on approach to software analytics, empirical software engineering research methods, and machine learning for software engineering. At the end of the course, the students will be able to: - Understand current literature in the area of software analytics and machine learning for software engineering - Apply software analytics techniques to extract actionable software engineering insights - Apply machine learning / deep learning algorithms to solve software engineering tasks	
Education Method	The course is a seminar, which means that we will be studying the literature in the area of software analytics and machine learning for software engineering. The course consists of the following education methods: - Self-study and presentation of papers - Development of software analytics tools and machine learning models to solve known / new problems To finish the course, students (in groups) will have to: - Study several papers (at least 10), which will be discussed during the lectures - Prepare and lead the discussion for 1 paper - Replicate existing work / propose new work The course may also feature guest lectures from top researchers in the area.	
Literature and Study Materials	Research papers are the main literature used in this course. We will share the resources with the students once the course starts.	
Prerequisites	Those are not strict prerequisites, but if you have followed those courses you will be better prepared for the Machine Learning for Software Engineering part of the course * CS4240 - Deep Learning * IN4325 - Information Retrieval	
Assessment	The final grade consists of the following items: * 80% - Results in a form of paper * 20% - Presentation of results The course does not have an exam and there will be no resit of any of the items above.	
Enrolment / Application	Each student who wants to take part in this course is *required* to: - register/enrol on Brightspace before the start of the course - participate in the first lecture of the course Failure to comply with these requirements may lead the student to be not allowed to take part in the course.	
Special Information	Due to resource constraints, this run of the course will only accept 20 students. If the number of enrolments is higher than 20, the selection procedure will be based on the student's motivation, as indicated by a short (1-2 paragraphs) motivation message that explains why the student would like to participate in this course.	
Tags	Project Research Methods Software Software Engineering	

IN4398	Advanced Practical IoT and Seminar	5
Responsible Instructor	Dr. R.R. Venkatesha Prasad	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	fundamental understanding of wireless communications, familiarity with wireless communications and embedded systems and knowledge of Android programming/Python/C++/Matlab.	
Parts	Each seminar: 2x 45 minutes (2 parts) + 10 minute break	
Summary	This course is an involved hands on course for self motivated students. Students work in a group of two (max 3) usually. Students are expected to have sufficient programming and hardware development skills. The course is project oriented thus students are expected to deliver a working model with demonstrations. Preference will be given to Q5 students. Q1 students are discouraged unless otherwise explicitly allowed. --- IMPORTANT NOTICE REGARDING ADJUSTMENTS DURING THIS COVID19 SITUATION The course will be offered ONLINE. Depending on the COVID19 situation there may be on-campus meetings/discussions [Safety of everyone is the highest priority]. --- This course is done in a group of two (max 3). Instructor/TA will help connecting a potential groupmate in the first class using Brightspace/Feedback Fruits. Activities: How to in Groups? ----------------------------- Part 1: Seminar We guide the groups to select a paper. Students need to prepare slides and deliver a seminar using online tools; it is a group activity. For each seminar all the students MUST be present. Absenting without permission will be taken seriously. Part 2: Project (1). Students can share the responsibilities if they can organize in such a way that one works on H/w and another on software so that they develop the project together. Meeting each other may be possible. However such face-to-face meetings should be done by strictly adhering to the COVID19 related instructions from the university/government. (2). Students may exchange the components/boards within the group but carefully adhering to 1.5m rule. The decision to do such cooperation is left to the judgement of the students and it is their own responsibility. Please note safety is first. For Students not in NL or not able to be in Delft: (3). Instructor/TA would be offering an opportunity to do a hardware project if students can buy simple hardware like Arduino boards/raspberry pi, etc. They should try to use the method as in (1) above. (4). Instructor/TA may offer Algorithm/simulations/android programmes/Contiki based networks or Open testbeds projects. In these cases, online group activities are possible. NOTE: In some extreme cases, Instructor/TA may allow one person project after assessing the situation and after discussing with the student. Supply of Hardware: ------------------- 1. Instructor/TA is able to provide take home components (limited numbers) like Arduino boards and some sensors. We may provide some other instruments/sensors/boards depending on the selected project. These could be collected from TA while adhering to sanitization and social distancing rules. 2. Students may also use their own components.	
Course Contents	Course will be composed of a series of seminars related to the broad topic of the Internet of Things. Students will present their results on investigations regarding the possible extension of the ideas presented in the assigned papers.	
Study Goals	To be able to design components of Internet of Things and showcase an application or product through an implementation of a project. Specifically, to be able to bring entrepreneurial aspect of the project and also to be able to evaluate the project in depth. To be able to criticize and assess system-level components of the Internet of Things environment discussed in the scientific literature.	
Education Method	Seminar will be composed of (i) seminar presentation on a selected research paper (from top journals/conferences) presented individually by students and, (ii) work on a research project. Students will be provided with a list of projects that will be assigned to them. Project will be summarized in the form of a written report (report must include critical analysis). Within a project any hardware/software platform can be used and demonstrated. User experience/study, where applicable, also needs to be executed. Selected paper needs to be critically evaluated and a proposal to extend the assigned paper will need to be presented in a form of a presentation. Paper extension should focus on a system level idea. Presentation skills, thinking and reflection abilities are looked into carefully. The teams are composed of two (or three) students generally. NOTE: The total amount of work would be on the higher side of 150 hours; since this is an advanced course, students are expected to already have very good knowledge of hardware platforms, coding, and design environment. In case of lacking in some of these skills, we expect students to acquire them outside this budgeted 150Hrs. However, the number of hours of workload mentioned here is ONLY a guide, the efforts depend on the project, goals, and the collaboration with the team member(s).	
Literature and Study Materials	This is a project based course. Thus, Internet and other appropriate manuals (for chipsets, etc.) would be useful. For papers to read and present, we expect students to look into: Infocom, Mobicom, IPSN, Sensys, Sensapp, Mobihoc conference papers. Journals: IEEE Trans on Networking, Trans on Mobile computing, JSAC, etc. ACM Trans on CPS, etc.	
Assessment	Part 1: Assessment based on presentation quality, slides, Q/A, constructive criticisms, ideas for improvements, etc.	

	Part 2: Project execution in a group, demonstration, Q/A and a report describing the outcome of the assigned project. Part 1: 30% of the whole marks; Part 2: 60% of the whole marks. In the assessment, a focus on the practicality and entrepreneurial aspect of the idea will be prevailing. A working model, demonstration, and a report, are expected. Individual Q/A after demonstrations will be part of the assessment. Part 3: Up to 10% marks would be awarded if the report is detailed above the minimum expected and the work is ready for a submission to a conference. Report will not have individual component, however, Q/A may have individual component. Please NOTE: There would be no resit since this is a hands-on course being executed in a team. disclaimer: Information may change depending on the developments around the coronavirus.
Tags	Circuits Group work Programming Project Software
Judgement	30 Marks for Seminar 60 Marks for project, demonstration, Q/A 10 Marks for making a paper that could be submitted as a manuscript for review to a conference/journal. Bonus marks are offered for excellence in various components during the course. That would add to the total marks not exceeding a max of 100.
maximum aantal deelnemers	Max 30 students, which translates to 10-15 groups (Groups consisting of 2 or 3 students).

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Free Elective Space 2021

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Quantum Computing 2021

AP3421	Fundamentals of Quantum Information	4
Responsible Instructor	Dr. L. di DiCarlo	
Instructor	Dr. D. Elkouss Coronas	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Knowledge of linear algebra, probability and statistics.	
Course Contents	Approximate syllabus: - quantum states, unitary operations, and measurements; - universal gate sets; - entanglement, Bell test; - basic quantum communication protocols; - basic algorithms and quantum algorithmic techniques; - basic quantum error correction; - simple physical implementations of qubits.	
Study Goals	Motivation: Quantum information is the future of computing and communication. Quantum computers offer exponential speedup over any classical computer. Similarly, quantum communication offers many advantages, including the ability to create secure encryption keys where security rests only on the laws of nature. Synopsis: This class will teach you the fundamental principles of quantum information. You will learn essential concepts that distinguish quantum from classical devices. You will learn about quantum bits and the quantum operations and measurements that can be performed on them. You will learn the basic techniques used in quantum algorithms, and examine basic examples of such algorithms. You will also take the first step in understanding how a quantum bit can be physically implemented. Aim: To learn the fundamental concepts underlying quantum computation and communication systems.	
Education Method	3 hours of lecture, 1 hour tutorial per week.	
Literature and Study Materials	The main reference textbook for the course is Nielsen and Chuang, Quantum Computation and Information, Cambridge University Press.	
Assessment	30% homework assignments, 10% in class quiz, 60% final exam. A minimum grade of 5.0 (unrounded) for the final exam is required to pass the course.	
Permitted Materials during Tests		
Continuing Courses	This course forms part of the curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information AP3421-PR Quantum Information Project CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation	

AP3432	Quantum Hardware 1 - Theoretical Concepts	4
Responsible Instructor	Prof.dr. B.M. Terhal	
Instructor	J. Borregaard	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Undergraduate electricity and magnetism; AP3421 Fundamentals of quantum information; Bachelor Quantum Mechanics	
Course Contents	Quantum hardware is what turns the novel concepts of quantum computation and communication into reality. The key challenge is to control, couple, transmit and read out the fragile state of quantum systems with great precision, and in a technologically viable way. Quantum Hardware I is focused on teaching theoretical physics concepts for understanding this Hamiltonian engineering challenge in various quantum hardware platforms. The material will be taught using example systems such as spin qubits (quantum dots or NV centers), superconducting, Majorana or trapped-ion qubits.	
Study Goals	Understand underpinnings of single-qubit and two-qubit gate dynamics, qubit measurement, Rabi oscillations, dephasing & relaxation times, dynamical decoupling. Understand various approximations to obtain effective Hamiltonian dynamics. Understand sources of noise and inaccuracy. Ability to work with Lindblad equations modelling noise. Ability to work with bosonic and fermionic systems.	
Education Method	Weekly lectures and tutorial exercise sessions with discussion of homework	
Course Relations	This course forms part of the curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation Other Related Courses: AP3112 Quantum Optics and Lasers	
Literature and Study Materials	Lecture notes Auxiliary textbook: Nielsen and Chuang Quantum computation and information, Cambridge University Press.	
Assessment	30% homework, 70% final written exam.	
Continuing Courses	AP3442 Quantum Hardware 2 - Experimental State of the Art AP3472 Modeling of Superconducting Devices AP3662 Special Topics in Quantum Technology	

AP3442	Quantum Hardware 2 - Experimental State of the Art	4
Responsible Instructor	Prof.dr.ir. L.M.K. Vandersypen	
Instructor	Prof.dr. W. Tittel	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Undergraduate electricity and magnetism; Quantum Hardware I (AP3432). AP3421 Fundamentals of quantum information AP3432 Quantum Hardware 1 - Theoretical Concepts	
Course Contents	Quantum hardware is what turns the novel concepts of quantum computation and communication into reality. The key challenge is to initialize, control, couple, transmit and read out the fragile stage of quantum systems with great precision and in a technologically viable way. While Quantum Hardware I is focused on teaching underpinning theoretical tools, Quantum Hardware II will give you an overview of the experimental state-of-the-art. You will learn about the most promising approaches for realizing quantum hardware, and critically assess the strengths and weaknesses of each approach. You will also get insight in the conceptual similarities and differences between the various technologies. Specifically, the course will cover general concepts and considerations of qubit hardware, trapped ions, superconducting circuits, quantum dots, impurities, cold atoms, photonic circuits, single-photon sources, single-photon detectors and quantum repeaters.	
Study Goals	- To acquire a good understanding of the requirements of quantum hardware for computing, communication and sensing, both at the conceptual level and at the practical level. - To acquire conceptual insight in the operation, opportunities, and challenges of various qubit realisations. - To obtain a good overview of the state-of-the-art.	
Education Method	In this course, lectures are combined with homework assignments as well as presentations of recent research papers that are to be studied at home.	
Literature and Study Materials	Reviews and research papers to be studied at home, material presented during lectures.	
Assessment	Grades will be established based on homework assignments (40%), paper presentations (25%), and a final exam (35%). The lowest homework grade does not count. This means you can miss one homework without having points deducted. Two missed homeworks will affect your grade, regardless of why you missed them. In unusual cases, a retake exam--which will be oral--will be offered by appointment.	
Continuing Courses	This course forms part of the QuTech Academy curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation	

CS4090	Quantum Communication and Cryptography	5
Responsible Instructor	Dr. S.D.C. Wehner	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear Algebra, Probability & Statistics, Q101 (Fundamentals of quantum information)	
Course Contents	This class will introduce you to the fascinating field of quantum communication! We will look at the state of the art of quantum networks, and explore techniques for building quantum repeaters that promise to deliver qubits over long distances. We also briefly look at one of the most famous application of quantum cryptography, quantum key distribution. Caution: 1. This class requires you to take "Fundamentals of Quantum Information" in Quarter 1 2. The focus of this class is presently on quantum communication, and we will only briefly look at quantum cryptography. As such, this class is not held in flipped classroom style in conjunction with edX QuCryptoX as in previous years.	
Study Goals	The student will acquire: A good understanding of the fundamental concepts of quantum communication Insight into the differences between classical and quantum communication and cryptography Skill set required to follow the remainder of the quantum curriculum (Q301 Quantum hardware and Q401 Quantum electronics)	
Education Method	Lectures and tutorials. If remote classes continue this fall, then recorded lectures and live discussion session	
Literature and Study Materials	Primary: Slides Review Articles Auxilliary: Nielsen and Chuang Quantum computation and information, Cambridge University Press.	
Assessment	Homework (70%) and Final Project (30%) disclaimer: information may change depending on the developments around the coronavirus.	
Tags	Abstract Adventurous Algorithmics Challenging Group Dynamics/Project Organisation Information & Communication Integrated Intensive Involved Lineair Algebra Mathematics Physics Quantum Signals Technology Telecommunication	

EE4575	Quantum Computing Architecture and Electronics - Fundamentals and state-of-the-art	5

Responsible Instructor	Dr. F. Sebastiano
Responsible Instructor	Dr. S. Feld
Contact Hours / Week x/x/x/x	0/0/4/0
Education Period	3
Start Education	3
Exam Period	3 4
Course Language	English
Expected prior knowledge	Students should have successfully completed the course Fundamentals of Quantum Information (AP3421). Basic knowledge of Matlab is also required.
Course Contents	The goal of this course is to introduce the students the overall system of a quantum computer, focusing on the classical hardware and software infrastructure required to build a quantum computer together with the quantum hardware. This course complements the course Quantum Hardware (AP3292) that covers the quantum components in a quantum computer by covering all classical hardware and software components. Covered topics: - Quantum computer architecture; quantum languages; compilers; QISA; microarchitecture. - Quantum simulator (QX simulator); - Quantum error correction: quantum error correction codes; encoding; logical operations. - Mapping of quantum circuits; placement of qubits; scheduling of operations; routing of qubits; - Quantum Classical Hybrid Computation; - Classical hardware for quantum computing: classical controller and quantum processor; qubit hardware and the need for cryogenic electronics; - Receiver and transmitter architectures for controlling qubits: frequency up/down-conversion, modulation schemes, frequency generation. - Hardware building blocks: amplifiers, analog-to-digital converters, digital-to-analog converters, digital processing, FPGA - Cryo-CMOS: device characteristics, state-of-the-art, design principles.
Study Goals	By the end of this course, students are able to: - Explain the different layers that are required to build a quantum computer. - Discuss the basic functioning of quantum error correction (QEC). - Carry out the mapping step of simple quantum circuits onto specific physical device layouts. - Interpret the results of different quantum circuits written in language QASM and executed using simulator QX, including the following: Basic quantum algorithms; Fundamental QEC codes; Surface code, especially for a Ninja star. - Identify and describe the different hardware blocks in the classical controller in a quantum computer. - Derive the specifications of the individual components in a classical controller form the requirements of the whole controller. - Compare different hardware blocks based on their specifications and can select the appropriate component for target functionality and specifications. - Describe the main features and advantages/disadvantages of the adoption of cryogenic electronics in general and cryo-CMOS specifically.
Education Method	Lectures.
Literature and Study Materials	- M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 2002. - Lecture slides and other material, such as scientific papers, will be distributed.
Assessment	Weekly homework and a final assignment. Final grade computed as follows: 50% homework; 50% final assignment. disclaimer: information may change depending on the developments around the coronavirus.

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Language Courses & Skills 2021

TPM018A	English Grammar for the University	2
Module Manager	Drs. D.W. Laponder	
Instructor	Drs. K.I. van der Linden	
Instructor	S.F. Johnson	
Instructor	Drs. M.A. Swennen	
Instructor	Drs. D.W. Laponder	
Responsible for assignments	Drs. D.W. Laponder	
Co-responsible for assignments	S.F. Johnson	
Contact Hours / Week x/x/x/x	2/2/2/2	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	none	
Course Language	English	
Course Contents	The course is designed for students who want to improve their English grammar in order to communicate more effectively. We use a university grammar that addresses the limitations many students face when interacting in English. The focus of this course is on improving your grammar range and accuracy to help you express complex thoughts more clearly.	
Study Goals	During this course, students will: 1. expand their basic understanding of sentence structure in academic discourse. 2. demonstrate an understanding of more complex grammatical structures 3. write a variety of sentence types 4. practise the grammar needed to write an academic text 5. demonstrate an understanding of grammar through text assignments 6. write paragraphs which use targeted grammar structures	
Education Method	It is a 7-week course with a weekly 90-minute class. If registration is lower than expected, the course may be offered as self-study in combination with a weekly consultation hour.	
Literature and Study Materials	Grammar for Academic Purposes 2 - Steve Marshall This textbook is available in e-format only (approx. 20) or a combined e-book and paper book (approx. 30). Details of how to obtain a copy will be given after enrolment.	
Assessment	Weekly assignments with set deadlines. If deadlines are not met, students may be asked to leave the course.	
Elective	Yes	
Category	BSc and MSc level	

TPM303A	Intermediate Writing in English for the University	2
Module Manager	Drs. K.I. van der Linden	
Instructor	Drs. K.I. van der Linden	
Instructor	S.F. Johnson	
Instructor	Drs. M.A. Swennen	
Instructor	Drs. D.W. Laponder	
Co-responsible for assignments	S.F. Johnson	
Contact Hours / Week x/x/x/x	2/2/2/2	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	none	
Course Language	English	
Course Contents	This writing course is a 7 week course with 1 lesson a week (2 hours) plus approx. 6 hours of homework assignments per week. The course focuses on the acquisition of necessary language skills in the areas of grammar and usage, as well as reading and writing at a level appropriate to university study. Students will develop skills that develop an idea with relevant support for that idea by structuring the text according to current writing conventions, using various transitions to link ideas and demonstrating a knowledge of grammar and punctuation. Students will edit their own texts and work with other students in peer-editing groups.	
Study Goals	The aim of the course is to develop language skills and writing skills needed for producing academic texts. This course will give the student an understanding of the conventions of academic writing in English and some of the grammatical structures common in academic texts.	
Education Method	Individual, pair and group work, combined with assignments to do at home. Feedback on course components is given by the course lecturer and other students.	
Practical Guide	This is a course for both BSc and MSc students who would like to expand and improve their English writing skills. When you have taken our Placement Test you will be recommended a language level. With this recommendation you can sign up for any of the courses offered at that level. For this course you need a recommendation to take courses at the Intermediate level.	
Books	Advance in Academic Writing 1 - Steve Marshall This textbook is available in e-format (approx. 32) or a combined e-book and paper book (approx. 52). Details of how to obtain a copy will be given on Brightspace after enrolment.	
Assessment	Students will be given a final writing assignment. In borderline cases teachers will take the final decision, based on the student's class participation and homework assignments.	
Enrolment / Application	Before enrolling in a group you must first take our Placement Test. For more information on this test please visit our website: https://www.tudelft.nl/tbm/over-de-faculteit/afdelingen/stafafdelingen/itav/onderwijs/english-unit/placement-test For this course you need a recommendation to take courses at intermediate level. After receiving your Placement Test results you can enrol via Brightspace. Please note that you must also choose a group to reserve a place in the class. Once you have enrolled in the course, go to collaboration and select the group that suits your schedule. You do not need to register in Osiris for this course.	
Special Information	Note that you are required to spend an additional 6 hours (approx.) per week doing homework assignments. Please make sure that your schedule allows you this time.	
Remarks	An 80% attendance rate is obligatory, which means you may miss one session. To keep your place in the group you must attend the first session. You can drop the course with no penalty before the second session. Be aware of your "commitment" from the start, as you will still receive a grade [NV: Niet Verschenen--Not Attending], in Osiris even if you stop coming to class. The course is offered every quarter. However, the course will only be offered if there is a sufficient number of participants.	
Elective	Yes	
Targetgroup	This course is available to BSc and MSc students	

TPM304A	Advanced Writing in English for the University	2
Module Manager	Drs. K.I. van der Linden	
Instructor	Drs. K.I. van der Linden	
Instructor	S.F. Johnson	
Instructor	Drs. M.A. Swennen	
Instructor	Drs. D.W. Laponder	
Co-responsible for assignments	S.F. Johnson	
Contact Hours / Week x/x/x/x	2/2/2/2	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	none	
Course Language	English	
Course Contents	This writing course is a 7 week course with 1 lesson a week (2 hours) plus approx. 6 hours of homework assignments per week. The course focuses on writing effective, well-structured and coherent academic texts in English. It looks at the writing process, including planning and structuring the text. Students will then develop paragraphs into longer academic texts and work on text coherence and cohesion. The course also pays attention to advanced academic vocabulary, sentence structure, punctuation and grammar.	
Study Goals	The aim of the course is to develop language skills and writing skills needed for producing academic texts. This course will give the student an understanding of the conventions of academic writing in English and some of the grammatical structures common in academic texts.	
Education Method	Individual, pair and group work, combined with assignments to do at home. Feedback on course components is given by the course lecturer and other students.	
Practical Guide	This is a course for both BSc and MSc students who already have a good understanding of academic writing, but who wish to expand and improve their English writing skills. When you have taken our Placement Test you will be recommended a language level. With this recommendation you can sign up for any of the courses offered at that level. For this course you need a recommendation to take courses at Advanced level.	
Books	Advance in Academic Writing 2 - Steve Marshall This textbook is available in e-format (approx. 30) or a combined e-book and paper book (approx. 50). Details of how to obtain a copy will be given on Brightspace after enrolment.	
Assessment	Students will be given a final writing assignment. In borderline cases teachers will take the final decision, based on the student's class participation and homework assignments.	
Enrolment / Application	Before enrolling in a group you must first take our Placement Test. For more information on this test please visit our website: https://www.tudelft.nl/tbm/over-de-faculteit/afdelingen/stafafdelingen/itav/onderwijs/english-unit/placement-test For this course you need a recommendation to take courses at Advanced level. After receiving your Placement Test results you can enroll via Brightspace. Please note that you must also choose a group to reserve a place in the class. Once you have enrolled in the course, go to collaboration and select the group that suits your schedule. You do not need to register in Osiris for this course.	
Special Information	Note that you are required to spend an additional 6 hours (approx.) per week doing homework assignments. Please make sure that your schedule allows you this time.	
Remarks	An 80% attendance rate is obligatory, which means you may miss one session. To keep your place in the group you must attend the first session. You can drop the course with no penalty before the second session. Be aware of your "commitment" from the start, as you will still receive a grade [NV: Niet Verschenen--Not Attending], in Osiris even if you stop coming to class. The course is offered every quarter. However, the course will only be offered if there is a sufficient number of participants.	
Elective	Yes	
Targetgroup	This course is available to BSc and MSc students	

TPM305A	Writing a Masters Thesis in English	2
Module Manager	Drs. D.W. Laponder	
Instructor	Drs. K.I. van der Linden	
Instructor	Drs. D.W. Laponder	
Instructor	S.F. Johnson	
Instructor	Drs. M.A. Swennen	
Responsible for assignments	Drs. D.W. Laponder	
Co-responsible for assignments	S.F. Johnson	
Contact Hours / Week x/x/x/x	x/x/x/x	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	none	
Course Language	English	
Course Contents	This course is designed for students who have started their MSc thesis or are about to start on their graduation project. The main focus is on the writing process, which means that matters such as grammar, academic style and precision of formulation are considered at length. Particular attention is also paid to the importance of structure and organising ideas. Aspects of the writing process will be backed up by weekly reading and writing assignments.	
Study Goals	The course is primarily designed for students working on their Masters thesis. As the name of the course indicates, the focus throughout is on writing. Sections of your academic writing will be corrected in considerable detail and you will be taught how to structure an essay, report or thesis. Plenty of tips will be given on ways of perfecting your English and invigorating your writing. Your systematic mistakes will be pointed out to you so that you can become critical about your own writing and go on to do your own editing.	
Education Method	Learning by doing and learning from lectures, corrections, peer review and tutor feedback.	
Books	Course materials will be made available through Brightspace.	
Assessment	Students are to attend a minimum of six classes, complete all weekly assignments and submit a final assignment of some 2,500 words (roughly 3 pages, not previously submitted) on a subject closely related to their graduation project / Master's thesis. A Pass grade will be awarded if all course requirements are met. The course will only run if enrolment numbers are sufficient.	
Elective	Yes	
Category	MSc level	

WM1115TU	Dutch Elementary 1	3
Module Manager	Drs. G. Hoezen	
Responsible for assignments	Drs. G. Hoezen	
Co-responsible for assignments	Dr. S.J. van Boxtel	
Contact Hours / Week x/x/x/x	3/3/3/3	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	1 2 3 4	
Course Language	Dutch English	
Required for	After this course, participants may pass on to beginners level (CEF A2) by doing the follow-up course WM1116TU or through self-study.	
Expected prior knowledge	None (The introduction lesson is in English. If you don't speak English, please contact us.) If you already know about 1000 words in Dutch or more, please contact us at delftsemethode@tudelft.nl	
Parts	speaking, listening tests, 2 writing assignments, 3 written tests	
Course Contents	[Please note: During the corona period the course format will be modified, e.g. online sessions instead of face to face classroom sessions, different testing formats] In the lessons, which will be based on texts on aspects of Dutch society that have to be learned before each lesson, it will be particularly speaking and listening skills that will be practised. During each lesson, knowledge of the texts will be tested with the aid of an automatic listening test programme. Participants will also have access to a multimedia learning, practising and testing programme available on the facultys network.	
Study Goals	To acquire basic Dutch language skills on CEF level A1.	
Education Method	Combination of conversation lessons, computer tests and self-study.	
Literature and Study Materials	Bondi Sciarone e.a., "Nederlands voor buitenlanders. De Delftse Methode". Course book. 5th revised edition. PLEASE NOTE: until August 2019 the 4th revised edition will be used. Check the website (www.tudelft.nl/dm) for the latest information.	
Assessment	The final grade is compiled of grades for written tests, listening tests, writing assignments to be performed during the course and a grade for presence during classes, combined with teacher's judgement of speaking skills . In case of a poor result, participants may sit a concluding test, which is a written test. To pass the course, one needs 75% for the concluding test + "pass/sufficent" for speaking.	
Permitted Materials during Tests	None (this also applies to the computer tests and writing assignments, which are part of the final grade).	
Enrolment / Application	For MSc-students, exchange students en BSc-students with an English programme: enrolment in TWO steps via Brightspace. STEP 1: pre-enrolment for the Introductory lesson and a introductory listening test on the first chapters through self-study. (Search for: 'Pre-enrollment for Introduction meeting & Introduction test' in Brightspace). After passing the test, go to STEP 2: enrolment for the actual course: 'WM1115TU Ducth Elementary 1'. For others (Phd, staff, ...): language desk, faculty TBM, room C0.010, or delftsemethode@tudelft.nl	
Special Information	Brightspace, www.dm.tudelft.nl or delftsemethode@tudelft.nl	
Remarks	The course provides extensive language training for foreign MSc and Exchange students who will learn how to express themselves well in simple Dutch, in everyday situations. Other categories (Staff, PhD researchers) may participate provided that they or the graduate school or their department pay for the course (for course fee see www.dm.tudelft.nl). Studyload per week (lessons + homework): 8-10 hours. More information: www.dm.tudelft.nl	
Elective	Yes	
Tags	Broad Diverse Small groups	
Targetgroup	anyone who knows less than 1000 words in Dutch and wants to learn Dutch and has enough time to prepare for the classes (8-10 hrs a week in total). People who already know some Dutch (appr. 1000 words or more) can contact delftsemethode@tudelft.nl for a level test.	
Self Test	If you want to do self-study, you can register for a written test + speaking test by sending an email to delftsemethode@tudelft.nl	
Category	MSc level	

WM1116TU	Dutch Elementary 2	3
Module Manager	Drs. G. Hoezen	
Responsible for assignments	Drs. G. Hoezen	
Co-responsible for assignments	Dr. S.J. van Boxtel	
Contact Hours / Week x/x/x/x	3/3/3/3	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	1 2 3 4	
Course Language	Dutch	
Required for	Intermediate 1	
Expected prior knowledge	Elementary 1(WM1115tu). Students who already acquired some knowledge of Dutch which is equivalent to that of the Elementary Course I level, i.e. CEF A1 (appr. 1000 words)should take a level test, to see if they can start with this course.	
Course Contents	[Please note: During the corona period the course format will be modified, e.g. online sessions instead of face to face classroom sessions, different testing formats] The lessons will be based on texts on aspects of Dutch life.These texts have to be learned before each lesson. It will be particularly speaking and listening skills that will be practised. During each lesson, knowledge of the texts will be tested with the aid of an automatic listening programme. After this course, participants will have attained beginners level (CEF A2), which means that they will be able to engage in simple discussions in Dutch on all kinds of everyday subjects.	
Study Goals	To obtain Dutch language proficiency at beginners level (CEF A2), especially for listening and speaking.	
Education Method	Combination of conversation lessons, computer tests and self-study.	
Literature and Study Materials	Bondi Sciarone e.a., "Nederlands voor buitenlanders. De Delftse Methode". Course book. 5th revised edition. PLEASE NOTE: until August 2019 the 4th revised edition will be used. Check the website (www.tudelft.nl/dm) for the latest information.	
Assessment	The final grade is the average of a grade for speaking skills (judged during classes) and a combined grade for presence during classes, grades for written tests, listening tests, and writing assignments to be performed during the course. In case of a poor result, participants may sit a concluding written and/or an oral exam.	
Enrolment / Application	Enrollment for Msc, Bsc and exchange students through Brightspace. For more information, contact your teacher or delftsemethode@tudelft.nl (language desk,ITAV).	
Remarks	The course provides extensive language training for foreign MSc and Exchange students who will learn how to express themselves well in simple Dutch, in everyday situations. Other categories (Staff, PhD researchers) may participate provided that they, the graduate school or their department pay for the course (for the course fee please visit our website www.dm.tudelft.nl). The course can be followed in the first, second, third or fourth quarter, provided there are enough participants. Study load: 8-10 hours per week.	
Elective	Yes	
Tags	Broad Diverse	
Category	MSc level	

WM1117TU	Dutch Intermediate 1	3
Module Manager	Drs. G. Hoezen	
Responsible for assignments	Drs. G. Hoezen	
Co-responsible for assignments	Dr. S.J. van Boxtel	
Contact Hours / Week x/x/x/x	3/3/3/3	
Education Period	1 2 3 4	
Start Education	1 2 3 4	
Exam Period	1 2 3 4	
Course Language	Dutch	
Required for	see Dutch page	
Expected prior knowledge	see Dutch page	
Course Contents	see Dutch page	
Study Goals	see Dutch page	
Education Method	see Dutch page	
Computer Use	see Dutch page	
Literature and Study Materials	see Dutch page	
Books	see Dutch page	
Prerequisites	see Dutch page	
Assessment	see Dutch page	
Permitted Materials during Tests	see Dutch page	
Enrolment / Application	Enrolment for Msc, Bsc and exchange students through Brightspace. For more information, contact your teacher or delftsemethode@tudelft.nl (language desk, ITAV).	
Special Information	see Dutch page	
Elective	Yes	
Tags	Broad Diverse	
Targetgroup	see Dutch page	
Category	MSc level	

WM1135TU	Advanced English for the University	3
Module Manager	Drs. K.I. van der Linden	
Instructor	Drs. K.I. van der Linden	
Instructor	S.F. Johnson	
Instructor	Drs. M.A. Swennen	
Responsible for assignments	Drs. K.I. van der Linden	
Co-responsible for assignments	S.F. Johnson	
Contact Hours / Week x/x/x/x	2/2/2/2	
Education Period	1 2 3 4	
Start Education	1 3	
Exam Period	none	
Course Language	English	
Course Contents	This course is given at C1 CEFR (Common European Framework of Reference) level. Advanced English for the University is the highest level integrated skills course offered by the English Unit within the ITAV institute at TPM. It is designed for those who already have a thorough working knowledge of English grammar and vocabulary but who still wish to further perfect their proficiency in order to become more adept in both writing and speaking. Time will be devoted to: Academic texts and writing conventions Academic vocabulary and collocations Grammar and punctuation Presentations and discussions Language learning strategies Advanced English for the University is a 14-week course with 1 lesson a week (2 hours) plus approx. 2 hours of homework assignments per week. The groups tend to be small (max. 15 participants) and the sessions are highly interactive. Attendance is mandatory. Students can miss a maximum of two lessons. Courses begin in the first week of September and in the first or second week of February . Please see the website for Placement Test dates and for any further course details: https://www.tudelft.nl/tbm/over-de-faculteit/afdelingen/stafafdelingen/itav/onderwijs/english-unit	
Study Goals	By the end of the course students should be familiar with different types of academic texts academic writing conventions academic vocabulary with a specific focus on collocations common grammatical errors punctuation conventions structural devices to improve coherence various language resources such as online dictionaries	
Education Method	Students will do in-class exercises completed either in small groups or in pairs. They will also do a group PowerPoint presentation and a discussion led in pairs. The feedback on course components will be given both by the lecturer and by fellow course participants.	
Literature and Study Materials	Course book:Cambridge Academic English - An integrated skills course for EAP (Student's Book; C1 Advanced). Authors: Martin Hewings and Craig Thaine Publisher: Cambridge University Press, 2012 ISBN: 978-0-521-16521-1	
Prerequisites	To qualify for the course students should have successfully completed Academic English 2 (previously known as English for Academic Purposes 3 - WM1101TU) and have gained a 7.0 or higher. Also, an Advanced English for the University recommendation following the completion of the Placement Test will also give direct access to the course. Please see in this connection: https://www.tudelft.nl/tbm/over-de-faculteit/afdelingen/stafafdelingen/itav/onderwijs/english-unit/placement-test	
Assessment	There is no final exam. Assessment is based on the written assignments and on the in-class discussions and presentations. The six writing assignments each account for 10% of your final grade while your grade for the discussion and your grade for the presentation each accounts for 20% of your final grade.	
Enrolment / Application	After receiving their Placement Test results students can enrol via Brightspace by first registering for the course and then selecting the group of their choice.	
Remarks	More information can be found on Brightspace	
Elective	Yes	
Targetgroup	BSc, MSc and PhD students are welcome	
Category	BSc and MSc level	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Projects 2021

TUD4040	Joint Interdisciplinary Project	15
Responsible Instructor	Prof.dr.ir. J. Hellendoorn	
Project Coordinator	Ir. B.J.E. de Bruin	
Contact Hours / Week x/x/x/x	x/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	Different, to be announced	
Course Language	English	
Summary	JIP consists of three sets of activities: 1. The project which takes place at a company. Students are responsible for the project work (collaboration, integration of different perspectives, the content quality of their work). They plan their activities via a scrum method, are challenged to realise field trips, consult with experts and realise the necessary research work. The team keeps a Scrum log, run a blog for interaction with the outside world and every team member keeps a personal development log. 2. Meeting Professionals lectures and workshops about specialised topics, by academic staff or senior professionals from the companies involved. 3. Plenary meetings and design reviews for all the teams, company coaches and academic staff. At the meetings the students present their intermediate resp. final outcomes and plans, and are provided with feedback.	
Course Contents	The aim of the Joint Interdisciplinary Project is to prepare students to contribute to solving impactful technological challenges. The projects not only demand good engineering working knowledge but also experience with interdisciplinary and systems theory, and both knowledge and mindsets of innovation and entrepreneurial behavior. The project brief is provided by renowned companies like Airbus, Arcadis, etc. Teams of interdisciplinary student teams guided by a company coach and offered academic and industry expertise, are invited to realise an innovative problem solution to a complex problem and contributing to the sustainable development goals.	
Study Goals	1. Cognitive abilities attributable to interdisciplinary learning Demonstrate the ability to engage in perspective-taking; Develop structural knowledge pertaining to the problem; Integrate knowledge and modes of thinking drawn from two or more disciplines; Produce an interdisciplinary understanding of complex problem or intellectual question. 2. Scientific and intellectual development Capable to analyze scientific and societal consequences (economic, social, cultural, environmental) of the innovation; 3. Research and design capabilities Demonstrate engineering skills: technical skills, interpreting results, creativity, usability for company/institute; Demonstrate that they are capable to independently apply relevant theory and/or knowledge to research and/or design; 4. Collaboration and communication in an interdisciplinary team Demonstrate behavioral competences and skills: taking initiative, responsibility, showing communication skills, independency, collaboration and the ability to respect different disciplines and adapt to different cultures); Show ability to write a technical report: structured/consistent, language proficient, with correct use of literature/references, use of figures/tables/equations, and has a concise format (30 pages); Present work performed in a structured way through an oral presentation to their peers and customer. 5. Self-adjustment and reflection capabilities Plan and control the project efficiently considering resources and methodology; Being able to reflect on personal functioning in an evaluation report: reflect on personal objectives, indicate personal strengths/weaknesses. Indicate future personal improvement, drawing conclusions for future career. Cognitive abilities attributable to interdisciplinary learning; The ability to integrate (scientific and practical technological)knowledge from different disciplines to solve complex problems Scientific and intellectual development The capacity to evaluate the ethical, scientific and societal consequences of the proposed innovation Research and design capabilities The ability to create reasonable and relevant research or design , according to the academic standards of the involved disciplines Collaboration and communication in an interdisciplinary team Demonstrate behavioural competences and skills relevant for teamwork and effective communication with different stakeholders Self-adjustment and reflection capabilities To carry out regular reflections on professional and personal development and being able to improve upon those reflections Understand contemporary and societal issues in their work.	
Education Method	Full-time project work in an interdisciplinary team of about five students. The project work is interspersed with some just in time seminars and workshops about specialized subjects, methods or practical situations that play an essential role in interdisciplinary project work.	
Assessment	The assessment criteria pertain to the process, product and presentation and are assessed at an individual and team level, during 3 review sessions at three levels of accomplishment (each review assesses at a different level): Interdisciplinary work Scientific reasoning and ethical mindset Innovation process Presentation and communication Academic staff, supported by the input from company coaches, grade the students during the Problem Statement, the Midterm and the Final Review. The final group mark for JIP is differentiated per person and is based on: The team presentations - 30% The Problem definition, progress report and final report 30% The individual contribution to the team 20% The final individual reflection report 5% The Blog 5%	
Remarks	Project Outcomes (product, model, system) are presented at: 1. Problem Statement Review 2. Midterm meeting 3. Final Review 4. Symposium presentation	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Thesis Project

IN5000	Final Project	45
Responsible Instructor	Dr.ir. A.R. Bidarra	
Responsible Instructor	Dr.ir. W.P. Brinkman	
Responsible Instructor	Dr. K.A. Hildebrandt	
Responsible Instructor	Dr. J.C. van Gemert	
Responsible Instructor	M. Finavaro Aniche	
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Contact Hours / Week x/x/x/x	x/x/x/x	
Education Period	1 2 3 4 Summer Holidays	
Exam Period	none	
Course Language	English	
Course Contents	IN5000 Final Project is the final part of the Master's degree programme. During this project, you will be required to demonstrate your ability to solve a research or engineering problem. The project must be carried out using the techniques of project management. You will begin by making a project plan in cooperation with your Masters thesis advisor. Several aspects of the project are defined within the plan, including the assignment, the frequency of interaction with the advisors, the milestones of the project and the resources and facilities offered by the faculty. You will be required to adhere to your plan throughout the project. It is obviously possible to adjust your plan under certain circumstances and after discussion with your daily supervisor. At the end of the project, you will submit your Masters thesis, which must be written in English, and make an oral presentation of your work to the Thesis Committee. The Thesis Committee will announce the final mark, which is based on the project performance, the thesis, the presentation and the subsequent discussion. More information about the graduation process: https://www.tudelft.nl/en/student/faculties/eemcs-student-portal/education/graduation-policy-msc/	
Study Goals	1. The student is able to design a research project: - The student is able to use, explain and justify adequate research and design methodologies; - The student is able to apply theory to the performed project; - The student is able to use techniques for interpretation and verification and bases his/her conclusions on results; - The student is able to do reliable work with scientific significance. 2. The student is able to execute a research project: - The student has a critical attitude towards his/her own results, literature and specialists; - The student makes an original contribution to the project; - The student takes initiative (together with the supervisor) to give his/her own input within the research project; - The student interacts sufficiently with peers and superiors; - The student is able to make and execute a project plan. 3. The student is able to write a research report: - The student is able to write a research report that shows sufficient coherence of content; - The students is able to structure the research report and sufficiently present the content (text and figures); - The student expresses argumentation using correct spelling and grammar; 4. The student is able to present and defend the research project: - The student is able to present the content using sufficient detail to support conclusions; - The student is able to logical structure the presentation and use visual aids; - The student is able to adequately formulate and express himself/herself as well as sufficiently address the audience; - The student is able to argument and answer the questions asked by the committee.	
Education Method	Project	
Prerequisites	Before starting the project, students must have completed at least 60 EC of the Master's degree programme and be in possession of a Thesis Entrance Permit (TEP). To be able to get a TEP, the individual exam program (IEP) should have been approved by the Board of Examiners. Note: In some cases, the thesis supervisor may impose additional conditions for starting this project.	
Assessment	The thesis committee assesses the thesis and the defense on the following criteria: - quality of work: novelty, volume, grasp, methodology, publishable; - personal performance: autonomy, planning, creativity, attitude; - quality of thesis report: clarity, organisation, argumentation; - oral presentation and defense: clarity, focus, relevance, discussion. More information on thesis grading: https://www.tudelft.nl/en/student/faculties/eemcs-student-portal/education/graduation-policy-msc/assessment/ The voting members of the thesis committee determine the final grade. The grade should reflect a weighted average of the four scores above, but need not to be an exact arithmetical mean. The final mark starts from 5 up to and 10. Marks ending in .5 may also be used. If the student shows excellence (is nominated for a 10) the chair of the thesis committee should consult the chair of the Board of Examiners, at least five working days in advance of the defense. The chair may advice to add an extra member to the thesis committee. The motivation for the grade at each of the four criteria as listed above is summarized on a form and signed by the chairman of the thesis committee. The candidate is given a short account of the assessment, either in private or in front of the audience. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Research Groups 2021

Introduction 1

Students do the Master Thesis project IN5000 under supervision of a CS research group from the INSY or ST department.

INSY - Intelligent Systems Department research groups:
1. CGV - Computer Graphics and Visualization
2. CYS - Cyber Security
3. II - Interactive Intelligence
4. MMC - Multimedia Computing
5. PRB - Pattern Recognition and Bioinformatics

ST - Software Technology Department research groups:
1. ALG - Algorithmics
2. DS - Distributed Systems
3. ENS - Embedded and Networked Systems [OLD NAME: Embedded Software (ES)]
4. PL - Programming Languages
5. SE - Software Engineering
6. WIS - Web Information Systems

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Algorithmics 2021	
Introduction 1	The Algorithms group designs and evaluates algorithms to solve problems in complex systems where decentralization, uncertainty, conflicting interests, and time constraints are major issues. Think of real-time balancing of energy supply and demand in communities of producers and consumers, or of coordinating schedules for service providers at airports in order to ensure that planes are cleaned at the right time and provided with fuel and food services. To design such algorithmic solutions we build upon fascinating fundamental scientific findings in computer science. In our group you learn how to design and use advanced algorithms using methods from planning and scheduling, algorithmic game theory, and sequential decision making under uncertainty. Also you will learn to implement these algorithms efficiently and to use the right methods to evaluate their performance. Once you have obtained this algorithmic expertise, you can participate in our groups research projects on topics such as smart grids, transportation systems, surveillance or maintenance. Quite a number of these projects involve industrial partners such as Alliander, NedTrain and Thales. A good example of a recent master thesis is Frits de Nijs thesis Project Scheduling: The Impact of Instance Structure on Heuristic Performance, available at our website (http://www.alg.ewi.tudelft.nl). This website also contains information about possible master thesis projects. To participate in a master thesis project you need to have completed at least our Advanced Algorithms course and two of our specialization courses.

CS4205	Evolutionary Algorithms	5
Responsible Instructor	Prof.dr. P.A.N. Bosman	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Must have the ability to program your own (extensions of) evolutionary algorithms (e.g., C, C++, Java). Further, although no particular high-level prior knowledge is expected, a basic understanding of key aspects in linear algebra (e.g., matrix multiplications, inversions, decompositions, etc.), probability theory and statistics (e.g., normal distribution, statistical hypothesis testing), algorithm design (e.g., efficient local search, heuristics), complexity theory (e.g., big-O notation, algorithm analysis), will be very helpful.	
Course Contents	In this course we consider a specific subfield of Artificial Intelligence: Evolutionary Algorithms (EAs). These algorithms, sometimes also identified as being part of the class of bio-inspired algorithms, have as a metaphor the concept of natural evolution, i.e., the mechanisms by which, the fittest individuals in a population survive, reproduce, and in doing so, over time, change to be better equipped to thrive in their environment. Initiated in the 60s and 70s of the 20th century, research on EAs has progressed immensely. Today, EAs are being used to solve real-world problems in many areas, e.g. to optimize the layout of electrical wind farms, to automatically create radiation therapy treatment plans, and to optimize the architectures of deep neural networks. This course covers a spectrum of topics in EAs, ranging from basic concepts to advanced, recent, and state-of-the-art research, and ranging from theoretical to applied. In particular, topics include genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, optimal mixing evolutionary algorithms, multi-objective optimization, and real-world applications. The course is planned to have 7 lectures and 2 practical assignments. The first practical assignment pertains to experimenting with already implemented EAs on predefined problems. The second practical assignment offers more freedom, allowing you, in a group, to build your own EA (this may vary depending on student numbers and other circumstances). disclaimer: information may change depending on the developments around the coronavirus.	
Study Goals	Upon successful completion of this course, students will be able to: 1) Explain the key concepts underlying the main streams in Evolutionary Algorithm (EA) research, with in particular genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, and optimal mixing evolutionary algorithms. 2) Explain key ingredients underlying the rationale of when these algorithms work and when they do not work. In particular: schema analysis and how the match between the search bias of an EA and the fitness landscape is influenced by aspects such as variable dependencies and multi-modality. 3) Name and explain key research lines along which state-of-the-art research in EAs is done to achieve more robust, efficient, and effective EAs. 4) Identify good opportunities for using EAs, or hybrid versions thereof, in practice. 5) Properly (scientifically) experiment with EAs as well as program your own.	
Education Method	7 Lectures 2 Lab projects disclaimer: information may change depending on the developments around the coronavirus.	
Literature and Study Materials	Papers and slides that will be made available.	
Assessment	The final grade is based on 60% written exam, 40% lab practical work. disclaimer: information may change depending on the developments around the coronavirus. In particular, there may be no written exam. In that case, there will likely be several smaller practical assignments and a large practical assignment at the end of the course. Both parts will then likely count for 50%. For the large assignment there will also be chance to resit this part by means of a repair option through an oral examination of the lecture contents.	
Permitted Materials during Tests	None	
Tags	Algorithmics Artificial intelligence Optimalisation	

CS4210-A	Algorithms for Intelligent Decision Making	5
Responsible Instructor	Dr. N. Yorke-Smith	
Instructor	Dr. E. Demirovi	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Expected prior knowledge	Recommended: IN4010: Artificial Intelligence Techniques, or equivalent; and/or IN4301: Advanced Algorithms, or equivalent Required: basic course(s) in algorithm design and analysis, logic and probability; basic programming (in Python)	
Course Contents	Decision making is at the centre of artificial intelligence. This course gives you practical skills on a solid theoretical base. The course looks at solving mathematical models of NP-hard discrete optimisation problems. These kinds of problems lie at the heart of AI techniques such as planning, machine learning and mechanism design, and more generally combinatorial optimisation. You will learn about a range of modelling techniques from boolean satisfiability to constraint programming, and how advanced solvers for these models work. The course has plenty of real-world case studies as well as theoretical results. Apply the skills you learn in this course by taking CS4210-B: Intelligent Decision Making Project in quarter 4!	
Study Goals	By the end of this course, you will be able to identify features of real-world combinatorial decision problems, and be able to model and design systems for simplified instances of these problems using boolean satisfiability, mixed integer programming, and constraint programming over finite and real domains. You will be able to explain how SAT, CP and LCG solvers work in some detail, and how MIP solvers work at a high level.	
Education Method	Lectures, homework exercises (optional), and programming assignments. The expected workload is: 30% lectures (including preparation for the exams) 40% homework exercises (optional) 30% programming assignments	
Literature and Study Materials	Provided on Brightspace	
Assessment	The final grade depends on the grades obtained for (a) programming assignments (2 in total) [30%] and (b) the exam [70%]. The final grade is computed from the unrounded grades for the components. The final grade for the programming assignment is a uniformally-weighted average of the unrounded grades obtained for the two assignments. Programming assignments can be completed by two students working together. The exam is graded on a scale from 1 to 10. A resit will be available for the exam. The result for the exam is determined by the maximum score obtained for the original exam and the resit. In order to pass the course, the rounded grade (after resit if applicable) for each part of the course must be at least 5.0, and the rounded final grade on the course must be at least 5.8. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Algorithmics Artificial intelligence Group work Modelling Optimalisation Programming Projects Small groups	

CS4210-B	Intelligent Decision Making Project	5
Responsible Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Dr. J.W. Böhmer	
Instructor	Dr. E. Demirovi	
Instructor	Dr. N. Yorke-Smith	
Instructor	Prof.dr. M.M. de Weerdt	
Contact Hours / Week x/x/x/x	0/0/0/1	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Theoretical knowledge regarding algorithms for decision making in Artificial Intelligence, obtained for instance by passing one of the following courses: - CS4210-A Algorithms for Intelligent Decision Making - CS4400 Deep Reinforcement Learning - IN4010(-12) Artificial Intelligence Techniques - IN4344 Advanced Algorithms.	
Course Contents	Decision making is at the centre of artificial intelligence. Building upon theoretical knowledge gained in other courses, students collaborate in small groups on a distinct research project per group, for instance on decision-making problems in transport, logistics or smart energy grids. Purely algorithmic challenges will also be provided. The research projects provide a good opportunity to learn about topics suitable for Masters projects in the Algorithmics section.	
Study Goals	After completing the Intelligent Decision Making Project course, the student is able to: 1. Apply algorithms for decision making to problem domains, and can compare and evaluate them. 2. Design and implement an extension of a decision-making algorithm. 3. Identify and discuss relevant topics in the research field of algorithms for intelligent decision making. 4. Describe and apply the appropriate research methodology. 5. Communicate his/her findings effectively.	
Education Method	A research project in a small group.	
Literature and Study Materials	Mainly survey papers and book chapters. Details are provided via Brightspace.	
Assessment	The assessment consists of the following items: 1. Quality of work of the research project (40%) 2. A scientific report of the research project (including peer review of a report) (20%) 3. Performance during the project (30%) 4. Oral presentation of the research project (10%) Only items 1 and 2 can be examined a second time. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Only a limited number of students can participate in this course. In order to be admitted, please submit a short motivation letter (max 200 words) via Brightspace. Attending the first lecture is compulsory.	
Tags	Artificial intelligence	
maximum aantal deelnemers	40	

CS4400	Deep Reinforcement Learning	5
Responsible Instructor	Dr. J.W. Böhmer	
Instructor	Prof.dr. M.T.J. Spaan	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students must have passed IN4010(-12) "Artificial Intelligence Techniques", or have acquired equivalent knowledge about: - basic probability theory, analysis and algebra - general machine learning methodology, e.g. regression - fully and partially observable Markov decision processes - tabular reinforcement learning methods, e.g. Q-learning - the exploration/exploitation trade-off, e.g. RMAX or UCB - multi-agent learning, e.g. centralized training and decentralized execution Students are encouraged to close any gaps in the above knowledge and to familiarize themselves with the Python/PyTorch deep-learning framework before the start of the course.	
Course Contents	This course will cover the breadth of modern model-free RL methods, discuss their limitations and introduce a variety of current research topics. In particular, we expect to cover the following: - deep learning methodology and architectures - stabilization of approximated value estimation - modern actor-critic methods - planning as inference - exploration with deep networks - offline reinforcement learning - deep multi-agent reinforcement learning - multi-task and meta learning	
Study Goals	After successful completion of this course, students - can list the strengths and limitations of modern deep RL approaches, - explain the underlying concepts of the discussed methods, and how they differ from each other, - can implement selected algorithms/architectures, and - can analyze a new task to decide which algorithms/architectures to apply.	
Education Method	The course will be taught in one lecture per week and the content will be solidified in homework, which will be presented in one mandatory tutorial per week.	
Assessment	The final grade will be 100% determined by a written exam at the end of Q3, with a resit possibility in Q4. To be eligible for the exam, students must attend weekly tutorials and hand in homework exercises. Homework will not be individually graded, but at least 75% of the answers must be of sufficient quality (in terms of time commitment, not necessarily correctness) to be eligible to take the exam.	
Maximum number of participants	As this is the first time the course will be taught, it will be restricted to 30 participants.	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Computer Graphics and Visualisation 2021	
Introduction 1	The focus of Computer Graphics and Visualization is the creation of visual content. This research field has many important applications in domains, such as architecture, health, geosciences, simulations, games, and movies. We work on many different topics in rendering, visualization of (scientific) information and modelling of 3D objects. Our goal is to develop new algorithms to generate models, represent and interpret data, and to find efficient solutions for display and interaction. One particularly important aspect are complex and large data sets, as they play an increasingly important role in many scientific, medical, entertainment, and engineering applications. In recent years, we offered several Master projects around topics related to our core fields: Game Technology, Geometry Processing, Interactive Visualization and Virtual Reality, Medical Visualization, Image Synthesis, and Rendering Techniques. We maintain an internal Master project page with example topics, but are also open to discuss additional possibilities. Currently, these options include work on scene compositing, physics-based deformations, stylization and artistic rendering, image processing, analysis of brain shape bio markers, visual analytics methods, geometric model fitting, stereo/multi-view rendering, and animation systems.

IN4089	Data Visualization	5
Responsible Instructor	T. Höllt	
Instructor	Prof.dr. E. Eisemann	
Contact Hours / Week x/x/x/x	0/2/0/0 & lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	Master course MKE/ST/DS	
Expected prior knowledge	IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required). The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.	
Course Contents	Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization, The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines. As a computer science course, affinity to algorithmic thinking and programing skills will be needed. Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.	
Study Goals	The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data. Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more. By the end of the course, you should be able to LO1: Discuss a large range of visualization techniques. LO2: Discuss a perception principle of visualization. LO3: Explain mathematical principles and algorithms of visualization techniques. LO4: Design suitable visualization systems for a given practical data analysis problem. LO5: Implement visualization systems for a given practical data analysis problem.	
Education Method	Lectures, practical assignments, self-study, and projects.	
Literature and Study Materials	Course slides, instructions for projects, and selected literature. Chapters from: Visualization Analysis and Design Author: Tamara Munzner CRC Press Visual Computing for Medicine 2nd Edition Theory, Algorithms, and Applications Authors: Bernhard Preim Charl Botha Morgan Kaufmann All available in electronic form via Brightspace or at TUDelft library.	
Assessment	The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Judgement	The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam. The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project. Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions. The project is evaluated based on the developed result, its documentation and presentation. Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.	

IN4152	3D Computer Graphics and Animation	5
Responsible Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. R. Marroquim	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Students that haven't followed any previous Computer Graphics courses (like TI1806) will be able to participate, but might have to invest some more time to catch up in the first lectures.	
Course Contents	Have you ever wondered how Toy Story was made, why the game Last of Us 2 looks so beautiful, or have you ever wanted to create your own graphics application or game? Then you should consider following this course. If not, then you should still follow it... maybe, you will become interested! In this course, you will get a good idea of Computer Graphics in general. The topic is of very high relevance for the industry and the research community and has numerous applications in different domains, such as scientific visualization, video games, simulators, special effects, animated movies and many more. Here, you will learn about basic algorithms, as well as modern techniques. We will address several topics: the principles of image synthesis, object representations, geometric and hierarchical transformations, graphics cards and the graphics pipeline, realistic rendering (including global illumination and effects, such as reflections), expressive rendering, physics simulations, rendering control (including previsualization systems used by professionals in the movie industry), and perceptual rendering, which relies on properties of the human visual system to enhance the quality of the images. Besides course sessions on the theory of Computer Graphics, some of the algorithms will also be reproduced in practice, and deepened during the final project.	
Study Goals	The course teaches computer graphics techniques on an advanced level. After the course the student is able to classify the different modeling, shading, and display techniques. The student can reproduce the basic mathematical and algorithmic notions associated with these concepts, can comment on the weak and strong points of these techniques, and can apply the core concepts within a graphics program in practice.	
Education Method	lectures, instructions, research papers, lab work	
Literature and Study Materials	Research Papers in domain of selected topics, lecture sheets, online sources, optional books (see below)	
Books	Fundamentals of Computer Graphics by Shirley et al. - CRC Press Real-time Rendering by Tomas Akenine-Möller, Eric Haines, Naty Hoffman - Peters, Wellesley Real-Time Shadows by Elmar Eisemann, Michael Schwarz, Ulf Assarsson, Michael Wimmer - Taylor & Francis Computer Graphics. Principles and Practice by James D. Foley, Andries VanDam, Steven K. Feiner - Addison Wesley	
Assessment	The course will be evaluated with two grades, a project grade, accounting for 60%, and a paper grade 40%. The project grade is the result of a project and its presentation that is building upon the assignments that are handed out (roughly) weekly during the duration of this course. The paper grade is the result of the presentation of a scientific paper and the development of an associated practical implementation. Details of both elements will be presented during the lecture. Both grades (project and paper) have to be at least a 5.0 and their weighted average should be 6.0 or higher after rounding (0.5 steps).	

IN4255	Geometric Data Processing	5
Responsible Instructor	Dr. K.A. Hildebrandt	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Basic knowledge in mathematics (linear algebra, calculus): TI1106M, TI1206M or comparable courses. Students who haven't followed any of these courses can follow the course, but should be willing to invest more time.	
Course Contents	Geometry processing is concerned with the representation, analysis, manipulation, and optimization of digital shapes. Thanks to the advances in 3D acquisition and manufacturing technologies (like 3D-Scanning and 3D-printing), the usage of geometric data is continuously increasing and an efficient processing of digital shapes plays an important role for a variety of applications in areas such as computer graphics, computer-aided design and engineering, medical imaging and surgery planning, architecture, and entertainment. In this course, we will study concepts and algorithms for creating, analyzing, editing and optimizing digital geometric shapes.	
Study Goals	After successfully completing this course, the student is able to: - describe the fundamental techniques used for representing, analyzing, processing and modeling digital 3D-shapes treated in the course and to explain the mathematical and algorithmic concepts associated with them - apply the learned mathematical concepts to solve basic geometric problems arising in geometric modeling applications - design algorithms that can solve simple geometric modeling tasks and evaluate the drawbacks, benefits and limitations of the proposed algorithms - implement the designed algorithms in a geometric modeling software framework	
Education Method	The course combines lectures, tutorials, practical project work, and homework assignments.	
Literature and Study Materials	References to textbooks and recent research and survey papers are given in the lectures.	
Assessment	The course will be assessed on two practical projects and two theoretical assignments. The course grade is a weighted average of the grades of the practical projects (60%) and the theoretical assignments (40%). Note that, there is a minimum grade of 5.0 for each assignment grade and the average grade for all components of at least a 5.8 in order to pass the course. Also, grades for individual assignments do not carry to the next year. Resubmission of modified coursework is only allowed for assignments that received a fail grade (<5.0). Overall resit grades will be capped to 6.0 disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4302TU	Building Serious Games	5
Responsible Instructor	Dr.ir. A.R. Bidarra	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2	
Course Language	English	
Expected prior knowledge	For CS students: programming experience with some object-oriented language; experience with graphics, AI and/or some game engine(s) is a plus. For all students: though not compulsory, it may be convenient to have followed the course SEN9235 (Game Design Project), which is taught in the first quarter.	
Course Contents	Project-based interdisciplinary course, open to MSc students of all faculties. The main goal of the project is to take students with varying talents, backgrounds, and perspectives and put them together to do what none of them could do alone: to design and implement a serious game aimed at being applied in a real-world setting (educational, social, training, health-related, etc.). The emphasis is both on constructively fulfilling the game requirements, and on deploying the adequate technology for that purpose. Assignments for this course will be provided by real-world end-users (e.g. companies or the Science Centre Delft), to whom the group will be reporting throughout the term of the project.	
Study Goals	At the end of the project, the student will demonstrate proficiency in the following aspects: o identifying and valuing the soft skills necessary to work in interdisciplinary teams o responsibly interacting within a team, integrating its members' varying talents and expertise o adapting with flexibility to the dynamic requirements of a complex external assignment o translating feedback received into proactive personal development steps Additionally, the CS student will demonstrate proficiency in the following specific aspects: o identifying, selecting and deploying the most adequate game technologies for the given serious game domain and constraints o deepening programming skills while building a complex and large software system in an agile context	
Education Method	Project: teams work intensively as a small game studio. Also a few plenary sessions and/or lectures (though less likely in 2021/22).	
Assessment	Project assessment will be based on a combination of: - (~50%) product grade: unique for the whole group, based on both the game itself and the required documentation; - (~45%) process grade (individual), including personal contribution, performance, attitude, and peer evaluation; - (5%) final presentation. The commissioner will be involved both as advisor and as assessor. The final documentation will include writing a scientific paper and actually submitting it to a conference on serious games and/or their application. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr. R. Marroquim	

IN4307	Medical Visualization	5
Responsible Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0 lectures & 0/4/0/0 lab.	
Education Period	1 2	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic knowledge of linear algebra, calculus and programming is needed. This course (IN4307) has been designed to complement the courses Advanced Image Processing (ET4283) and Medical Imaging (AP2231TUD). However, these two courses are NOT pre-requisites.	
Course Contents	Theory and practice (Notice project extends to Q2) of medical visualization. This includes the following aspects: data acquisition basics, clinical practice; image processing, e.g., filtering, segmentation and measuremen; medical volume visualization; illustrative visualization; advanced visualization for complex modalities; interaction techniques for medical data; advanced applications.	
Study Goals	By the end of the course, you should be able to LO1: Explain medical visualization algorithms and their applicability to medical problems. LO2: Discuss the advantages and disadvantage of medical visualization algorithms. LO3: Build a medical visualization system for a given problem: a. Discuss a suitable visualization for a given medical problem. b. Implement the most suitable solution. c. Judge the performance of the implemented solution.	
Education Method	The course will be based on a combination of lectures and practical assignments. A final project will be developed in Q2	
Literature and Study Materials	Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications Bernhard Preim and Charl P. Botha (not mandatory)	
Assessment	The evaluation will be based on - a written (or oral if the number of students allows) exam (40%) - assignments during the semester (10%) - a final project (50%) The final project will be done during the 2nd quarter. Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The assignments will consist of small programming exercises and open questions, as preparation for the final project. The practical sessions will provide time for working on the assignments with assistance. The deliverables will be program code and/or answers to questions. The final project will be the design and implementation of a visualization system for a given medical problem. The final project will be carried out in teams. The deliverables for the final project will be a report (paper), the results (e.g., code) and a short video presenting the project (i.e. screencast). The written exam will be arranged at the end of the first quarter. You are allowed to have the slides and material of the course during the exam. No computer or laptop is allowed. The exam has a resit. The project will have a resit if the exam (NOT the resit exam) has been passed with a mark of 7.5 or higher and it will be on an individual basis. The project resit is not automatic and must be initiated by you within two weeks of the grades being published. Resit of a project will mean starting a new project. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	Notes and written material. No computers.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4310	Seminar Computer Graphics	5
Responsible Instructor	Dr.ir. A.R. Bidarra	
Instructor	Dr. R. Marroquim	
Instructor	Prof.dr. E. Eisemann	
Instructor	Dr. K.A. Hildebrandt	
Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	One of the CS core courses (IN4086 Data Visualization, and IN4152 3D Computer Graphics and Animation), and at least one of the Computer Graphics specialization courses (IN4255 Geometric Modeling, IN4302 Building Serious Games, and IN4307 Medical Visualization) are expected as prior knowledge.	
Course Contents	In this seminar you work on a selection of recent topics and results in one of the areas of Computer Graphics.	
Study Goals	To obtain in-depth knowledge about an advanced topic within Computer Graphics, in particular in rendering, game technology, visualization, appearance capture, animation or geometric modeling. The seminar may be used as a preparation for an MSc thesis topic. The student will acquire practical skills in reading, presenting, explaining, and discussing scientific papers, as well as writing scientific papers.	
Education Method	This course has the format of a student seminar. Students will prepare a scientific presentation of a recent research paper. The presentation goes in-depth and covers the papers strengths and weaknesses. After each presentation, a research discussions takes place and will be held in the plenary colloquium sessions. Students will participate in a scientific discussion of some of the presented papers. Finally, each student will realize a particular aspect of the chosen paper in form of an implementation, which will be presented in a dedicated session and described in a short document (1 page + figures) following the structure of a scientific article.	
Literature and Study Materials	Recent research papers about the selected topic.	
Assessment	The course will be assessed based on several components: - Presentation of a recent scientific paper (50%) - Implementation component reproducing an aspect of the paper (40%, including short report and presentation) - Preparation of questions for some of the presented research papers (10%) The course is passed if all components have at least a grade of 5.0 and the weighted average is a 6 (rounded on 0.5). disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Cyber Security 2021	
Introduction 1	Our society critically depends on cyber space for almost everything, including banking, transport & logistics, air travel, energy, telecommunications, flood defences, health care, email, social networks, and even warfare. The consequences of cyber security failures could be disastrous and the demand for cyber security specialists is therefore high and rising. The Cyber Security (CYS) Section is within the Intelligent Systems Department in the Faculty of Electrical Engineering, Computer Science and Mathematics (EEMCS). It has strong connections with the ICT Section in the Faculty of Technology, Policy, and Management (TPM), with the 3TU Federation, with the Hague Security Delta, and with companies like Fox-IT. Hence, there are plenty of opportunities to explore the multi-disciplinary aspects of cyber security and to contribute to cutting-edge research. The research focus of the CYS Section is on the following topics. Computing with Encrypted Data - Secure Information Sharing - Homomorphic Encryption - Lightweight Cryptography Data Analytics, Machine Learning - Privacy Preserving Data Mining - Automated Reverse Engineering - Botnet Detection - Monitoring and Analytics Applications of (Quantum) Information Theory - Information Theoretic Security and Privacy - Quantum Information, Computation, Crypto, Error Correction (within QuTech)

CS4035	Cyber Data Analytics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	The course provides theoretical and practical background for applying data analytics in the field of cyber security. Cyber data analytics is a huge field with a great diversity of techniques and applications. The course is centered on a selection of seven such techniques: learning from imbalanced data; behavioral profiling and anomaly detection; sequential data mining; data stream mining; learning from software data; adversarial machine learning; and privacy-aware data mining; Anomaly detection is one of the main topics in cyber security. Specific difficulties that the student will learn to handle are the huge amounts of data and the large number of false positives. Behavioral profiling applies to both people and software processes. Different techniques will be taught to construct profiles from software logs. While building such profiles, care should be taken to not infringe upon the privacy of individuals the data is collected from. Finally, attackers will modify their behavior in order to avoid being detected, a cyber data analytics engineer tries to make their models/profiles robust against such modifications.	
Study Goals	The student will be able to: Apply machine learning to real data Understand and modify machine learning algorithms Learn models from time series Detect anomalies in multidimensional time-series Use distributed processing to speed up machine learning Learn models from data streams with limited memory Learn sequential models Use machine learning for fingerprinting and profiling Preserve the privacy of data owners while learning models Learn robust models that can detect evasive attackers Use machine learning to detect fraud, attacks, and botnets	
Education Method	There will be two lectures for each of the seven topics, and 3 large lab assignments on fraud detection, anomaly detection, and behavioral profiling, and 1 smaller lab on adversarial robustness. There is no exam. Teams of two students will work on these assignments which contain both individual and collaborative components. Deadlines are strict as peer-review will be used to both learn of other possible solutions, provide feedback, and get initial estimates on the obtained grade.	
Assessment	3 large lab assignments on fraud detection (30%), anomaly detection (30%), and behavioral profiling (30%), and 1 smaller lab on adversarial robustness (10%).	

CS4090	Quantum Communication and Cryptography	5

Responsible Instructor
Dr. S.D.C. Wehner

Contact Hours / Week x/x/x/x
0/4/0/0

Education Period
2

Start Education
2

Exam Period
2
3

Course Language
English

Expected prior knowledge
Linear Algebra, Probability & Statistics, Q101 (Fundamentals of quantum information)

Course Contents

This class will introduce you to the fascinating field of quantum communication!

We will look at the state of the art of quantum networks, and explore techniques for building quantum repeaters that promise to deliver qubits over long distances. We also briefly look at one of the most famous application of quantum cryptography, quantum key distribution.

Caution:
1. This class requires you to take "Fundamentals of Quantum Information" in Quarter 1

2. The focus of this class is presently on quantum communication, and we will only briefly look at quantum cryptography. As such, this class is not held in flipped classroom style in conjunction with edX QuCryptoX as in previous years.

Study Goals
The student will acquire:
A good understanding of the fundamental concepts of quantum communication
Insight into the differences between classical and quantum communication and cryptography
Skill set required to follow the remainder of the quantum curriculum (Q301 Quantum hardware and Q401 Quantum electronics)

Education Method
Lectures and tutorials. If remote classes continue this fall, then recorded lectures and live discussion session

Literature and Study Materials
Primary:
Slides
Review Articles

Auxilliary:
Nielsen and Chuang Quantum computation and information, Cambridge University Press.

Assessment
Homework (70%) and Final Project (30%)

disclaimer: information may change depending on the developments around the coronavirus.

Tags
Abstract
Adventurous
Algorithmics
Challenging
Group Dynamics/Project Organisation
Information & Communication
Integrated
Intensive
Involved
Lineair Algebra
Mathematics
Physics
Quantum
Signals
Technology
Telecommunication

CS4120	Seminar Science and Methods in Cyber security	5
Responsible Instructor	Dr. S. Picek	
Instructor	Dr. M.P.M. Franssen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	This seminar course Cyber Security covers the following topics: (i) an introduction to the philosophy of (classical and design) science, (ii) the art of writing a scientific research proposal, (iii) an overview of useful and relevant scientific methods, (iv) introduction to scientific writing (of a paper and of a MSc thesis).	
Study Goals	1. Getting a basic knowledge and understanding of what science entails and how scientific knowledge is being created 2. Getting knowledge and understanding of relevant scientific methods applicable in the field of Cyber Security 3. Getting knowledge, understanding and skills for writing a research proposal related to the creation of a MSc thesis 4. Getting knowledge and understanding on how to execute a scientific article and MSc thesis 5. Getting knowledge and understanding of how to execute a literature review.	
Education Method	Lecturers supported by the execution of mostly individual assignments. Attendance of participants in this course is mandatory.	
Assessment	Final grade will be based on a weighted average of various scores including (i) presence and level of participation (10%), (ii) quality of the research proposal to be written and presented (60%), (iii) grades for assignments (paper evaluation, paper rewrite, essay questions/written exam) (30%). No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Research Methods	

CS4150	Systems Security	5
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Language based Security (CS4105) and Security and Cryptography (IN4191) and Network Security (ET4397IN) and a Bachelor level Operating Systems course. The topics below should be covered to a good bachelor level. To allow students to assess their level of knowledge, they can have a short oral interview.	
Course Contents	IoT security, Hardware, Countermeasures, Covert channels, Secure System Engineering	
Study Goals	The student will acquire: An appreciation of the security architecture of computer systems Detailed knowledge of the security of a specific operating system Skills in exploiting vulnerabilities of computer systems Skills in developing counter measures against exploits	
Education Method	2 hours per week lectures, 4 hours per week lab	
Assessment	50% lab work (with automated testing for scalability) and 50% written open book examination. disclaimer: information may change depending on the developments around the coronavirus.	

CS4185	Capstone Cyber Security	5
Responsible Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	2/1/1/1	
Education Period	1 2 3 4	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	Please note that the course has 3 parts. All parts must be done within 12 months from the start of part 1 and in the following order: 1. Q1 & Q2: Capstone Social Skills https://www.4tu.nl/cybsec/en/course-program/cps/ 2. Q3 & Q4: Capstone Entrepreneurial skills https://www.4tu.nl/cybsec/en/course-program/cpe/ 3. Q5*: Capstone Business skills https://www.4tu.nl/cybsec/en/course-program/cpb/ *) Be aware that if you start this course in 2020-2021 (part 1 and 2), then part 3 will start at the beginning of the next academic year 2021-2022. Attendance is mandatory and there is no (chance of) reparation if you miss any session. NOTE: the study guide information may change depending on the developments around the coronavirus	
Study Goals	see the links provided	
Education Method	see the links provided	
Assessment	see the links provided disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	15 participants max, 4TU.CybSec students have priority.	
Co-Instructor	Dr.ing. V.E. Scholten	

CS4265	Computer and Network Security: Advanced Topics	5
Responsible Instructor	Prof.dr. M. Conti	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	*DISCLAIMER: study guide information may change depending on the developments around the coronavirus.* The course takes the form of seminars based on a selection of scientific papers (that either have had a strong impact on security today, or explore novel ideas that may be important in the future). The list of topics can be found in the brightspace Topics and Papers module. For each topic there is a primary paper, and possibly other additional papers. All the students are required to read all primary papers and be able to competently discuss the material in class. Each student will be responsible for presenting one lecture (based on one of the primary paper including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion in the class. 48 hours before each lecture each student must upload on a shared repository at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). This is intended to be an interactive class: class participation is strongly recommended (and will play a role in the grading criteria). Sleeping during the class is optional, but not recommended.	
Study Goals	This course is about learning to study, analize, do and criticize research in cybersecurity. This will be done by being exposed to actual research topics and scientific papers and discussing things together.	
Education Method	Studying, presenting and discussing recent research results in Computer and Network Security.	
Assessment	Presentation + Class Discussion + Written Report + Oral Exam (please refer to the Judgement field for more details) disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Judgement	The final grade will be made up of four components: 25% the presentation done by the student during the course: each student will be responsible for presenting one topic (based on the corresponding primary paper, including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion (Q&A) in the class. This component is based on following criteria: (15%) Layout and Graphics (30%) Content (20%) Organization (20%) Presentation (15%) Q&A. 25% for the active participation in Q&A sessions during the course: 48 hours before each lecture each student must submit (via email, to both the lecturer and the teaching assistants) at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). The students should actively participate in the discussion of the topics in the 10 minutes Q&A session for each presented topic. 25% for content and quality of the final essay: At the end of the course, each student must write a 5-page long essay about one of the topics that has been discussed in class, or another topic agreed with the lecturer. The topic and the structure of the essay must be agreed with the lecturer. The essay might include some implementation prototype or experiments/simulations to evaluate/support the claim in the paper (in case this is a significant part of the essay, two students can agree with the lecturer to work together). If the student cannot attend the lectures, an alternative work (e.g. a longer essay) must be agreed with the lecturer. 25% for the oral presentation of the essay: during the oral exam, the student is asked to give a 15-minute presentation to the lecturer and the teaching assistants about the essay (presenting with slides is highly recommended). During the oral presentation, students can also be asked questions about other topics of the course. This component is based on following criteria: (30%) Style (20%) Originality (50%) Organization (clarity in your argumentation, coherence between assumptions and conclusions, logical organization, evidence to support claims)	
Co-Instructor	Ir. S.E. Verwer	

CS4430	Network Security	5
Responsible Instructor	Dr.ing. A. Zarras	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	The course provides an overview of the most important concepts, methods, and best practices in computer and network security. In this course, students will obtain the knowledge and hands-on experience to secure networking and communication systems. The course's primary focus will be on technologies, protocols, attacks, and defenses. More precisely, starting from a review of common vulnerabilities and attack scenarios, the course will discuss the fundamentals of security engineering and their application in system design, review tools and methods to assess and test communication infrastructure from a security perspective. As a result, students will gain theoretical knowledge and hands-on experience in network attacks and defense methods. Knowledge activation and the transfer from conceptual understanding towards practical experience will be further facilitated by students implementing their own attack or defense tools on selected topics, as well as conducting measurements on the effectiveness of attack and defense schemes.	
Study Goals	See course contents.	
Education Method	Lectures, Labs, and Project.	
Assessment	Assignments and Project.	

IN4191	Security and Cryptography	5
Responsible Instructor	Z. Erkin	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	UT-201500042 Privacy Enhancing Technologies (Q4)	
Expected prior knowledge	Basic understanding on the following is suggested. -Probability and statistics -Programming skills	
Course Contents	Motivation: Computers are now found in every layer of society, and information is being communicated and processed automatically on a large scale. Examples include medical and financial files, automatic banking, video-phones, pay-tv, teleshopping and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy. Synopsis: Security and cryptography are essential components of any digital system. In this course, the fundamentals of secure data storage and transportation of information are described. In particular, classical (e.g. Caesar, Vigenere) and modern encryption schemes (RSA, DES, AES, Elliptic curves) are described along with their mathematical background such as number theory. Methods for authentication, data integrity and digital signatures are discussed in detail, as these are the main components of many security architectures. The course also investigates more advanced topics such as zero-knowledge proofs and secret sharing schemes. Aim: It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security and privacy, as well as is familiar with present applications. Learning outcomes: The goal is to make students familiar with the basic concepts applied cryptography, including classical cryptography and modern secret key and public key cryptography. In particular, the students will acquire A sound understanding of the notion of security An understanding of the confidentiality, integrity and authenticity needs of the society Understand the role of cryptographic primitives including the differences between symmetric and asymmetric cryptography, the role of hash functions, digital signatures and PKI Understand the advanced topics in cryptography needed for the modern society with untrustworthy entities Among others things, the following topics are covered: -Classical systems -Information theoretic security -Definition of Security notions -Symmetric encryption (e.g. DES, AES) -Asymmetric encryption (RSA, Elliptic Curves) -Hash functions -Random number generation -Key Management -Digital Signatures, -*Secret Sharing. (if time permits) -*Zero Knowledge proofs (if time permits)	
Study Goals	It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security, as well as is familiar with present applications. Through assignments, students are expected to have the chance to work on the topics covered in the lectures.	
Education Method	Lectures, assignments and weekly exercises. Attention: This course requires full effort of 140 hours. Even more, if you lack the background (probability and modular arithmetic) Planned Workload: Lectures: 28 x 45minutes sessions, total 22 hours Practice session: 7 x 90 min. total 12 hours Assignments: 3 x 20 hour, total 60 hours Weekly study: 7 x 4 hours, total 28 hours Exam preparation: 20 hours Exam: 3 hours	
Literature and Study Materials	Cryptography made simple, Nigel P. Smart, 2nd Edition, Springer, 2016 (PDF Available Online) Handouts of lectures	
Assessment	Written exam(70%) + mandatory assignments (30%). Passing grade for the written exam and assignments is required. Please refer to CS regulations for further details. There is NO reparation for assignments. Points from previous years cannot be transferred. NOTE: the study guide information may change depending on the developments around the COVID-19 pandemic	
Exam Hours	In case of in person examination at campus: The exam is closed book. A cheat sheet of size A4, hand written is allowed for the written exam. Name and student number has to be present on each side. In case of remote exam from home: open book: textbook, slides and self-made notes only. randomised/customised exam.	

Permitted Materials during Tests	Only non-scientific calculators.

IN4253ET	"Hacking Lab"-Applied Security Analysis	5
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Necessary background differs per student project, see first lecture or contact instructors for details	
Course Contents	The security of computer and telecommunication systems is becoming an increasing concern. In this course, we will review the current state of the art on security research and gain practical experience in assessing the security and vulnerabilities of communication systems. Engineers are typically taught to focus on performance, correctness, scalability, and maintainability when building communication and information processing systems. However, an additional set of design principles are required to achieve security. In this course, we discuss security principles, common pitfalls and vulnerabilities. The weekly lectures provide an introduction into security research, with a focus on real-world security, privacy-enhancing technology and common security pitfalls. Each student participates in a "Hack Project", with a group of one to four students. Students can select between a wide range of available Hack Project outlines within the first week. The goal may be to evaluate the security of a real-world IT system, developing a proof-of-concept exposing a vulnerability or focussed on preserving privacy in a post-Snowden world. Students may propose their own Hack Project based on their background knowledge and skills. Such Hack Projects need to be approved and shaped together with the instructor. Example of possible outlined hardware-oriented projects are: development of a wifi tracker, programing an FPGA system to break passwords, assess the security of RFID cards, or to transparently intercept Ethernet traffic. Concrete software projects are: hacking Bitcoin, improving the TOR anonymity protocol and create Android-based tools for human rights activists in Iran, Egypt and Russia, reprogramming neural networks attacks. Each Hack Project is documented with a written report. This can be in the form of a 6-8 page IEEE-style scientific article or a traditional more lengthy report. All results, experiences and findings are presented to the entire class in the last week of the course. Hack Projects also report their progress several times during the course, after the weekly lectures.	
Study Goals	After this course, the student will have a thorough knowledge of security in real-world systems, and will be able to explore the literature on this topic independently. The student will be aware of the poor state of security in real-world computer systems. The student can explain the common pitfalls, why these known failures still occur and reasons behind the poor state of security in general.	
Education Method	Lectures, student presentations, written final report and active participation. Attendance and active participation during lectures is mandatory. This sadly means telelecturing is not possible.	
Literature and Study Materials	Customize literature lists and study materials are provided per project topic	
Assessment	The final class grade is composed of several partial grades. Partial grades are given for the written Hack Project report (60%), final presentation of result (10%), presentation of ongoing project progress (20%), participation in discussions, overall quality of the practical work and class attendance (10%). Students are required to obtain a passing grade on all partial grades. Attendance to lectures is mandatory. No final written exam. No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	If there is an unexpected high demand for this course, then enrollment will be based on past performance in relevant courses.	

UT-191612680	Computer Ethics	5
Responsible Instructor	mr. J.M. Kooijman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	none	
Course Language	English	
Course Contents	https://www.4tu.nl/cybsec/en/course-program/coe/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=191612680&collegejaar=2020&taal=en	
Assessment	One written take-home exam at the end of the quartile, which counts 40% of the grade, and one group assignment (essay), which counts for the remaining 60%. The resit policy is that students are allowed to resit either the assignment or the exam once, irrespective of their initial grade.	
Enrolment / Application	Only for students enrolled in the 4TU cybersecurity master track of computer science. Requirements: - An UT student guest account is required. See https://www.utwente.nl/onderwijs/student-services/procedures-services/aanmelding-inschrijving/bijvakker-of-kies-op-maat/#registratie-bijvakker-of-kies-op-maat - The exam will be held at the TU Delft; therefore students must register here: https://www.tudelft.nl/studenten/onderwijs/tentamens/aanmelden-voor-tentamens/	

UT-192110940	Secure Data Management	5
Responsible Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	2/0/0/0+ assignments	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/sdm/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=192110940&collegejaar=2020&taal=en	

UT-201100022	Cyber Crime Science	5
Responsible Instructor	Dr. R.S. van Wegberg	
Contact Hours / Week x/x/x/x	0,0,2,1+project	
Education Period	3 4	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	0,0,2,1+project	

UT-201500038	E-Law	5
Responsible Instructor	mr. J.M. Kooijman	
Module Manager	mr.dr L.C.P. Broos	
Instructor	mr. J.M. Kooijman	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	See the website https://www.4tu.nl/cybsec/en/course-program/	
Enrolment / Application	Only for students enrolled in the 4TU cybersecurity master track of computer science. Requirements: - A UT student guest account is required. See https://www.utwente.nl/onderwijs/student-services/procedures-services/aanmelding-inschrijving/bijvakker-of-kies-op-maat/#registratie-bijvakker-of-kies-op-maat - The exam will be held at the TU Delft; therefore students must register here: https://www.tudelft.nl/studenten/onderwijs/tentamens/aanmelden-voor-tentamens/	

UT-201500039	Security Verification	5
Contact Hours / Week x/x/x/x	2/0/0/0+ assignments	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/sev/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=201500039&collegejaar=2020&taal=en	

UT-201500040	Introduction to Biometrics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/2/0/0+ project	
Education Period	2	
Start Education	2	
Exam Period	none	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/bio/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=201500040&collegejaar=2020&taal=en	

UT-201500041	Cyber Security Management	5
Responsible Instructor	Prof.dr.ir. P.H.A.J.M. van Gelder	
Module Manager	Prof.dr.ir. P.H.A.J.M. van Gelder	
Contact Hours / Week x/x/x/x	0/2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/csm/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=201500041&collegejaar=2019&taal=en	
Enrolment / Application	Only for students enrolled in the 4TU cybersecurity master track of computer science. Requirements: - A UT student guest account is required. See https://www.utwente.nl/onderwijs/student-services/procedures-services/aanmelding-inschrijving/bijvakker-of-kies-op-maat/#registratie-bijvakker-of-kies-op-maat - The exam will be held at the TU Delft; therefore students must register here: https://www.tudelft.nl/studenten/onderwijs/tentamens/aanmelden-voor-tentamens/	

UT-201500042	Privacy Enhancing Technologies	5
Responsible Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/0/2+ assignments	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	IN4191 Security and Cryptography Course	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/pet/ anonymous communication; identity management; anonymous credentials; anonymity systems; mix networks; onion routing; database privacy; k-anonymity; differential privacy; other probabilistic approaches; private data processing; secure multiparty computation; (fully/somewhat) homomorphic encryption; garbled circuits; secret sharing; privacy-preserving clustering; private recommender systems; private smart metering; privacy-preserving biometrics.	
Study Goals	Concepts like the Internet of Things or Big Data inherently utilize massive amounts of data containing private information collected and stored by websites, sensors, monitoring systems, auditing systems, and so on. Examples include electronic records in health care systems and location information in ubiquitous computing applications. But how can we protect the privacy of participating users while at the same time enable effective sharing and utilization of the distributed data? There are several dimensions in the area of privacy, ranging from technical and juridical to societal and economical. While we will touch upon all these different aspects in the course, we will focus on the technical dimension. We will explore potential techniques for building new platforms, services, and tools that protect users' privacy. The study of promising component technologies ranging from advances in anonymous communication and identity management to theoretic tools like differential privacy and cryptography will be the core of this course. Learning objectives: Good understanding of privacy in the Internet of Things Ability to analyse and evaluate anonymity mechanisms, both for anonymous communication and for database privacy Ability to apply and analyse the concept of secure multiparty computation to protect privacy in different application domains Gain hands-on experience with different privacy-enhancing technologies	
Education Method	Lectures (joint lecturing), lectures slides and articles.	
Literature and Study Materials	Lecture Slides, essential reading material provided with the lectures (articles and tutorials)	
Assessment	70% written, closed book, exam and 30% assignments. Passing grade for the written exam is required. In case of remote exam from home: open book exam, with random oral checks. NOTE: the study guide information may change depending on the developments around the coronavirus	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Cyber Security/SERG 2021

CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5

Responsible Instructor	Ir. S.E. Verwer
Responsible Instructor	Dr. A. Panichella
Contact Hours / Week x/x/x/x	0/0/4/0
Education Period	3
Start Education	3
Exam Period	3 4
Course Language	English
Course Contents	Software is one of the most complex artifacts of mankind has ever created, but complexity is the enemy of correctness. Modern software testing and validation tools use a multitude of techniques geared toward correct computer code, most of these are base on artificial intelligence. In this course, we study these techniques in details, specifically we will understand and implement: Execution monitoring and taint analysis Branch distance computation Hill-climbing and genetic algorithms Concrete and symbolic (concolic) execution Active state machine learning Genetic programming The goal is to better understand and test software using artificial intelligence. Using the taught techniques you will be able to automatically: Discover which code is reachable Find (security) bugs in software Write tests that cover all reachable code Reverse engineer a code's functionality Patch code to remove bugs and failing tests
Study Goals	The student will: Understand modern AI techniques for software testing. Be able to implement several such techniques from scratch: - smart fuzzing (probing software with input to find crashes/bugs), - symbolic execution (using logic to construct inputs that trigger specific code branches), - fault localization (given that a program fails, find the line of code responsible for the failure), and - automated program repair (using a patch library and genetic programming to improve code) Be able to apply this technology to locate bugs in real-world software implementations.
Education Method	The main part of the course will consist of 3 lab assignments covering the theory (fuzzing&tainting, symbolic execution, automated program repair), and one lab assignment for the application to real software. The students will implement the taught techniques from scratch in the first 3 assignments, which will be scored with a pass/fail. All three assignments need to be passed to complete the course. The final lab will contain a recap from the first three assignments and an application of a state-of-the-art tool on real software. The final lab will be graded and be the final course grade. There will be instruction sessions where students can work on their assignment and ask the teachers for assistance.
Assessment	First three lab assignments (pass/fail). Final lab (100%).
Tags	Artificial intelligence Software

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Distributed Systems 2021	
Introduction 1	The Distributed Systems group concentrates on the modeling, the design, the implementation, and the analysis of parallel and distributed systems and algorithms. This research is fundamental in that we aim at the development and evaluation of generic methods and techniques, and application-driven in that the research is motivated by application areas. Most of our research is experimental: we try to build prototypes of systems, preferably used in the real world, to demonstrate the quality of the proposed solutions. The main research areas of the group are P2P systems and online social networks, large-scale reputation systems and crowdsourcing systems, grids and clouds, and multicore architectures and parallel programming. The teaching of the group includes MSc courses on High-Performance Computing, Parallel Algorithms, Distributed Algorithms, and the Hacking Lab. We have excellent computer and lab facilities for its experimental research. We have a 32-node cluster computer (DAS-4; 32x2x4 cores, 18TB storage, 10 Gbps networking). Additionally, we have a Bitcoin-driven infrastructure for anonymous online purchases of equipment. A good example of a recent thesis is the work of Risto J.H. Tanaskoski titled Anonymous HD video streaming. This project created the first scalable privacy-enhancing technology capable of supporting video streaming. An operational and Internet-deployed prototype evolved in various stages. First, basic onion routing technology was implemented. Next, the codebase was extended with the Tor routing protocol for relaying. Finally, it was extended with support for Libtorrent and crypto-readyness was implemented. Performance measurements showed that the download speed exceeded 5 Mbyte per second. Therefore, provided there are sufficient user-donations of proxy bandwidth this thesis created an initial solution for anonymous streaming from existing Bittorrent swarms.

CS4160	Blockchain Engineering	5
Responsible Instructor	Dr.ir. J.A. Pouwelse	
Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	In this course you will learn all aspects of blockchain technology, including tamper-proof data structures, digital identities, transitive trust, fault tolerance, distributed consensus, smart contracts and applications. Ledgers and blockchains are an emerging technology with the potential to radically improve financial transactions, supply-chain flows, transactions in general, and distributed databases. The first three weeks of the course will provide a fast-paced introduction to Bitcoin, Ethereum, and TrustChain developed by TUDelft itself. The main component in this course is a team-based complex engineering project. This course is designed for computer scientists to understand blockchain technology and to produce significant hands-on experience. To provide a deep understanding of blockchain technology and understand why it is special you need to experience first-hand how it operates at a detailed technical level. Students design, implement, and test their own independent project in teams of 3-5 students. Students can choose from a pool of possible project ideas. This course requires you to like software engineering. Topics covered: -Blockchain basics and evolution Bitcoin 1st generation, smart contract generation, future 3rd generation (trust or trust in math) -identity and transitive trust Authentication and security primitives, tamper-proof identities, trust models, MITM attacks, Sybil attacks, and TrustChain by TUDelft -Consensus models Proof-of-work, permissioned, Proof-of-stake, Corda no-global-consensus, TUDelft bottom-up fast consensus model -Smart Contract pro/con encrypted data, Bitcoin scripts, Etherium execution model, Hyperledger + Docker issues, Corda Jar file approach, Tezos difficult to use, powerful technology, vision of the future: trusted verified execution -Markets and exchanges Disruption by open markets, winner-takes-all, and multi-sided market platforms, Uber, Airbnb, 22 years of eBay, Silk Road, honesty among drug dealers, the role of trust in markets, P2P exchange markets -Decentralized Autonomous Organization, novel method to collaborate and organise any economic activity Within this 2021 edition "the Delft DAO" will be prominently featured. TUDelft achieved a world-first in DAO research. We devised a full end-to-end proof-of-principle of a DAO which is capable of 0) near unbounded scalability 1) controlling money 2) democratic decision making and 3) continuous sustained self-evolution. This course provides you with the knowledge to work with this advanced technology. After this course you will have a firm grasp on the current operational blockchain-based systems, realistic view of real-world applications that may be built on top of ledger technology. You will be able to reason and discuss the open challenges and questions that still need to be resolved. This course is a key course for distributed systems students.	
Study Goals	After this course students are able to design and engineer complex blockchain-based systems. Students are able to describe blockchain technology, the various consensus model, smart contracts, markets, and relation to existing database technology. Student are able to setup a new architecture for blockchain applications.	
Education Method	This course consists of four 2-hour lectures. Each lecture is followed by a 4-hour homework period in the same week focused on understanding the background material. In week 1 you will form teams and initiate work on your blockchain engineering project. A list of projects to select from will be provided at the start of this course.	
Literature and Study Materials	Online course textbook: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Narayanan, Bonneau, Felten, Miller and Goldfeder.	
Prerequisites	It is highly recommended to follow this course (see remarks): Security and Cryptography (Q1) Distributed Algorithms (Q2)	
Assessment	The final grade reflects the quality of your work and team cooperation. This course has a minimal amount of formalities. You will write down your project results in a single-page report, IEEE style. You will be graded on your open source efforts located on Github and single-page report. Your grade will be expressed on a scale of 0 to 10. (resits or repair options are not offered for this course) Covid-19 disclaimer: the assessment and course format could be altered at any time !!!	
Remarks	This class has a limited capacity (50). If there is a larger number of enrollments than the capacity of the class, students will be assigned to their preferred blockchain engineering project based on their background, engineering experience level, and match to the course goals. Students who followed Security and Cryptography (Q1) and are also enrolled in Distributed Algorithms (Q2) will have priority for placement. Mathematics students are exempts from this, if they can show some minimal software development experience (e.g. Github profile). Finally, students with a Grade Point Average of 8.0 or higher are eligible for the challenging scientific projects, resulting in a research paper. These project receive intense guidance, but have no capacity limits.	

CS4215	Quantitative Performance Evaluation for Computing Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	Todays computing systems become ever complex, due to the rapid development of hardware and software technology. It is challenging to design and run computing systems that guarantee users performance requirements in a resource efficient way. Various quantitative methods are applied to capture such complex system dynamics and predict metrics of interests, from the designing phase of the systems to the runtime performance, e.g., job response times and system anomaly. To optimize the performance of computing systems, a deep understanding on those methods and their applications on the system design are essential. Having practical hand-on experience on designing experiments, deriving models, and validating results with benchmark systems will prepare students to tackle challenges of real world computing systems.	
Study Goals	LO1. Design full/fractional factorial experiments for multi-variate regression analysis, e.g., finding critical parameters for deep learning clusters LO2. Apply queueing theory to analyse and predict run-time performance of applications, e.g., the average response times of on-line ML training service LO3. Apply machine learning models to analyse and predict the system dependability, e.g,root cause analysis for machine failure. LO4. Conduct experiments to profile applications and extract their workload parameters on real systems, e.g., e.g., deep learning clusters LO5. Develop resource management policies and validate them on real computing systems, e.g., deep learning clusters	
Education Method	Lectures: 7 weeks X 2-4h Practical: Derive, validate and evaluate performance models and resource management strategies for a chosen system via homework and group project. Multiple types of computing and network systems can be chosen from. Deliverables include a report and group presentation.	
Books	1) Performance Modeling and Design of Computer Systems: Queuing Theory in Action by Mor. Harchol-Balter 2) Design and Analysis of Experiments by Douglas C. Montgomery, 3) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics.	
Assessment	Homework (40%): 2 individual homework Group project (60%): group project report and presentation disclaimer: information may change depending on the developments around the coronavirus.	

CS4285	Seminar: Decentralized Systems	5
Responsible Instructor	Dr. S. Roos	
Instructor	Prof.dr. J.S. Rellermeyer	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	Systems with one central party enable large-scale surveillance, suffer from a lack of reliability, and open the door to censorship and easy manipulation of public opinion with severe consequences for, e.g., elections. Decentralization avoids or at least mitigate these problems but comes with a number of challenges such as maintaining high performance and legal compliance. In this course, we first discuss advantages and disadvantages of decentralization. Afterwards, we group decentralized systems into three levels: 1) systems that add a decentralized component to centralized systems to enhance privacy (e.g., anonymity networks); 2) systems that have no central servers but still fully depend on the standard Internet architecture to work (e.g., blockchains), 3) systems that have no central servers and do not (only) use the Internet architecture (e.g., ad-hoc networking). Students will gain an overview of approaches and concrete systems in all three categories. They will further evaluate the introduced systems with regard to security, privacy, and performance. Course topics include Consequences and challenges of decentralization Onion routing, its implementation in Tor, and challenges faced by Tor Techniques to resist censorship and their impact on performance Methods for structuring overlay networks Anonymous and censorship-resistant overlay networks such as Freenet Censorship-resistant blockchain networks Methods for communicating without (directly) connecting to the Internet	
Study Goals	Define key concepts in the field of decentralization Describe the positive and negative impacts of decentralization on security, privacy, and performance of applications Explain and assess the key algorithms of deployed decentralized systems Apply mathematical proofs, simulations, or real-world measurements to evaluate decentralized systems	
Education Method	1. Lectures: 7 weeks X 2h 2. Paper reviews: Students read papers and come up with a survey. The course is blended and knowledge of the content of the papers is necessary to follow the subsequent lecture in-depth. 3. Practical: Students have two homework assignments where they evaluate a given decentralized system with regard to its privacy, security, or performance. 4. Presentation: The course contains a presentation of the conducted work	
Assessment	1. Paper reviews/survey (50%): The student will survey a set of papers 2. Practicals (40%): There will be two homework assignments, each worth 20%, to be submitted in week 4 and 8. Students have three weeks to complete the homework. 3. Presentation (10%): The student presents their work during the last week. There will be no classical exam and no resit for the practical assignments. Partial grades do not carry to the next year.	

CS4290	Seminar on Distributed Machine Learning Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	Machine learning systems are often conventionally designed for centralized processing in that they first collect data from distributed sources and then execute algorithms on a single server. Due to the limited scalability of processing large amount of data and the long latency delay, there is a strong demand for a paradigm shift to distributed or decentralized ML systems which execute ML algorithms on multiple and in some cases even geographically dispersed nodes. The aim of this seminar course is to let students learn how to design and build distributed ML systems via paper reading, presentation, discussion, and project prototyping. We provide a broad overview on the design of the state-of-the-art distributed ML systems, with a strong focus on the scalability, resource efficiency, data requirements, and robustness of the solutions. We will present an array of methodologies and techniques that can efficiently scale ML analysis to a large number of distributed nodes against all operation conditions, e.g., system failures and malicious attacks. The specific course topics are listed below. The course materials will be based on a mixture of classic and recently published papers. For each topic, the basic concepts and technology landscape will be first provided and then two state-of-the art of papers will be presented and discussed by students. We offer a testbed of a distributed (deep) ML system in which students are encouraged to apply different techniques to jointly improve its scalability and resilience. Course topics include Overview of distributed machine learning systems Performance and scalability of state-of-the-art systems Acceleration of ML workloads Slim distributed ML systems on small data Robust deep learning systems Federated machine learning systems	
Study Goals	Students are able to argue and reason about distributed ML from a systems perspective. Students understand the behavior and tradeoffs of distributed ML in terms of performance and scalability. Students can estimate the importance of data inputs via different techniques, i.e., core set and decomposition methods, for distributed ML systems. Students understand data poison attacks and design defense strategy for distributed ML systems. Students can analyze the state-of-the art federated machine learning systems and design the failure-resilient communication protocols. Students are able to design and implement methods and techniques for making distributed ML systems more efficient.	
Education Method	Lectures: 7 weeks X 2h Papers: one paper presentation, two paper reviews, and paper discussion. Practical: apply system and algorithmic optimization techniques learned in the lecture to improve the performance of distributed machine learning systems, e.g., image recognition on CIFAR 10. The testbed environment, learning algorithms, and dataset will be given. Deliverables include git commit of functioning code and a report summarizing the contribution	
Assessment	Paper presentation by group (10%): each group of 2 to 3 students needs to choose from a given set a papers to present (15 minutes) and lead the discussion (10 minutes). Paper reviews by individual (30%): each student needs to write three reviews of papers assigned from a given set and those two papers have to be different from the paper for the presented. Each review will account for 10% of the grade. Questionnaires by individual (0%): each student needs to hand in a list of questions at the beginning of the lectures that have paper presentation from other students. This is not graded. Individual project (60%): The project is collaborative among the entire class and competitive as a whole group. The objective is to continuously improve the performance of the given distributed ML system. The students need to hand in a final project report in style of a short scientific paper, stating their individual contribution to the overall system performance. disclaimer: information may change depending on the developments around the coronavirus.	

IN4150	Distributed Algorithms	6
Responsible Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Knowledge of Computer Networks (CSE1405) and Operating Systems (CSE2430) is useful as background for understanding this course.	
Course Contents	Introduction to distributed algorithms; notions of time and ordering of events; distributed algorithms for message ordering, detecting global states, termination detection, deadlock detection, mutual exclusion, election, minimum-weight spanning trees, fault tolerance, consensus, and agreement; blockchain technology and its relation with consensus.	
Study Goals	Understand the main fundamental problems in distributed systems Understand the most important distributed algorithms that solve these problems Be able to reason about the execution of distributed algorithms Be able to program distributed algorithms Be able to select and summarize relevant literature on distributed algorithms	
Education Method	Lectures and lab work executed in groups of two students	
Literature and Study Materials	Lecture notes and lecture slides (available on Brightspace)	
Assessment	One paper summary and an in-person, on-campus written exam (closed book), which together are an integrated whole. The grade of this exam, which includes the grade of the paper summary (with a weight of 25%), is the grade for the course. In case in-person, on-campus written exams are not possible because of COVID-19, the written exam will be replaced by individual online oral exams.	
Permitted Materials during Tests	None except the list of algorithms	
Remarks	Lab work is 40 hrs.	

IN4391	Distributed Systems	5
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Starting with the mid-1990s, computing is undergoing a revolution, in which collections of independent computers appear to users as a single, albeit distributed, computing system. Motivated by the increase in the computation capacity of consumer computers, by the commoditization of server-grade machines, and by the advent of the Internet, the distributed computing paradigm has permeated all fields using computers. Current distributed computing applications range from the consumer social networks, peer-to-peer file-sharing, and massively multiplayer online games; to scientific computing using Big Data and distributed sensors; and to engineering fields and industrial control systems. This course focuses on the systems aspects of distributed computing. Specific, contemporary distributed systems are used as illustrative examples to discuss system design and non-functional requirements.	
Study Goals	1. Explain the objectives and functions of distributed computing systems. 2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems. 3. Describe the architecture and operation of distributed computing systems. 4. Explain how distributed computing systems can process user workloads. 5. Explain how distributed computing systems can detect and correct faults and errors. 6. Implement complex operations of modern distributed computing systems in realistic scenarios. 7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)	
Education Method	Lectures: This class uses a partially flipped classroom setting. Students are required to prepare parts of the material through self-study to be able to follow the in-class discussion. This involves, e.g., reading scientific papers. Practical: Designing, implementing, and evaluating a complete distributed system in groups, based on existing research work. Multiple topics are given to choose from. Deliverables include the code and a report of max. 10 pages.	
Literature and Study Materials	Textbook: Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Principles and Paradigms (2nd Edition), Prentice Hall, 2006. The textbook introduces the student to the traditional theory of distributed systems. Additional material: Several relevant research articles introduce the student to the classic literature as well as the latest advances on the topic.	
Assessment	Written exam (closed-book, open questions) during the exam period. A resit for the exam is offered in the following exam period. Practical project assessed based on the code, a presentation, and the report. This course uses gamification. Points can be collected through the practical project (max 4000 points) and the final exam (max 6000 points). The final grade is determined proportional to the 10000 points total. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Embedded and Networked Systems 2021

CS4055	High Performance Data Networking	5
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Basic understanding of networking and programming (ideally Python).	
Course Contents	The Internet has become of critical importance to society. However, the large size of networks and abundance of protocols have made network management very complex. The novel concept of network programmability addresses this complexity and has resulted in a paradigm shift in how networks are (or can be) operated. The high-performance data networking course is an advanced networking course that will introduce you to the concept of network programmability and which treats fundamental networking concepts like Quality of Service and network resilience.	
Study Goals	The learning objectives of this course are twofold: (1) The student should gain knowledge of the treated networking technologies. (2) The student should be able to apply and work with the programmable network technologies in a network emulator (Mininet).	
Education Method	Approximately 50% of the course will consist of lectures and selfstudy and 50% focuses on (homework) exercises and instruction classes.	
Literature and Study Materials	Slides and a reader containing the exercise material.	
Assessment	The final assessment will be based on an exam that covers both the theory from the slides as well as the content from the reader.	
Remarks	Disclaimer: The information about CS4055 (including its assessment) may change depending on the developments around the coronavirus.	

CS4140ES	Embedded Systems Laboratory	5
Responsible Instructor	Prof.dr. K.G. Langendoen	
Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/4 Lectures + 0/0/0/4 Lab	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	MUST have C programming skills. Students who have taken the CSE2425 Emb. Software course automatically qualify, others will have to pass an on-line ACCEPTANCE test.	
Course Contents	This highly multi-disciplinary course comes with a lab project where teams of 4 students each will have to develop an embedded control unit for a tethered electrical model quad rotor aerial vehicle (the Quadrupel drone), in order to provide stabilization such that it can hover and (ideally!) fly, with only limited user control (one joystick). The control algorithm (which is given) must be mapped onto a home-brew PCB holding a modern RF SoC interfacing a sensor module and the motor controllers. The students will be exposed to simple physics, signal processing, sensors (gyros, accelerometers), actuators (motors, servos), basic control principles, and, of course, embedded software (C) which is the programming language to be used in order to develop the control system. The project work (including written report) covers the entire duration of the course period, and will take approximately 128 hours, of which 32 hours are spent at the lab facilities. This is a core course of the Masters in Embedded Systems.	
Study Goals	Student is acquainted with real-time programming in an embedded context, along with a basic understanding of embedded systems, real-time communication, sensor data processing, actuator control, control theory, and simulation. Moreover, the student has had exposure to integrating the various multidisciplinary aspects at the system level.	
Education Method	Lectures (8*2hrs), lab work (8*4hrs), coding@home (8*12hrs), report (8hrs), so on average 2 days per week	
Literature and Study Materials	Lecture notes + Website	
Assessment	Lab. project (75%) + written report (25%), no exam, no resit disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	The capacity is limited and -as this is a compulsory course for ES students- they get preference over other MSc students.	

CS4425	Visible Light Communication & Sensing	5
Responsible Instructor	M.A. Zuñiga Zamalloa	
Instructor	Q. Wang	
Contact Hours / Week x/x/x/x	0/0/0/4 Lectures 0/0/0/4 Lab	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Requirement: Programming experience This requirement is equivalent to having passed the course TI 1206 in our first year Bachelor curriculum "Object Oriented Programming". If you have not taken this course, experience with programming in C, Python and/or Java is sufficient. Some basic knowledge in wireless communication would help (modulation and demodulation), but it is not required because these concepts will be refreshed during the lectures.	
Course Contents	Nowadays, half of the worlds population (plus billions of things) connect to the Internet using mainly wireless technologies. Our ever-increasing demand for wireless is crowding the radio frequency spectrum: we need more bandwidth. To overcome this challenge, a new generation of devices have started to use the visible light spectrum to transmit data. This new paradigm is promising because the visible light spectrum is empty and free. In this course, you will be introduced to the general field of visible light communication and sensing (VLCS). You will learn about its history, current developments and future opportunities. The course will cover theory and practice. First, you will learn the main modulation techniques used to transmit data with light, and then, you will use that knowledge to create your own VLCS system. The course will also cover the latest research done on this topic at various universities around the world. A wonderful introduction about the potential of Visible Light Communication is given in this TED talk: https://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb?language=en#t-4148 This course includes 8 lectures, two lab projects, and one final main project. There are two lectures per week during the first four weeks. After that, there will be only lab sessions to monitor the progress of the main project. The 8 lectures are: Lecture 1: Introduction & Hardware Lecture 2: Physical Layer: Pulse-Based Modulation Lecture 3: Physical Layer: Advanced Modulation Lecture 4: Network Layer Lecture 5: Camera-based communication Lecture 6: Backscattering Communication Lecture 7: Sensing Applications Lecture 8: SoA, Standardization & Commercialization	
Study Goals	By the end of this course, you will achieve the following Learning Objectives (LOs): LO1: Illustrate the principles of Visible Light Communication (VLC) systems: a. Understand the characteristics of VLC devices such as LEDs and photodiodes. b. Formulate the principles of line-of-sight and non-line-of-sight VLC links. c. Elaborate on pulse-based and advanced VLC modulation schemes. d. Explain and innovate upon the mechanisms behind VLC networking systems. LO2: Employ the theoretical and experimental training on VLC systems to: a. Analyze LED-to-camera and backscattering VLC systems. b. Investigate novel visible light systems. LO3: Interpret the standardization, commercialization, and the state-of-the-art of VLC. LO4: Build and demonstrate a VLCS system with off-the-shelf devices.	
Education Method	Lectures + Labs + Final Project The lectures, labs and final project cover the entire duration of the course period and will take approximately 120 hours, of which 16 hours are spent on lectures, 44 hours on the two labs (22 hours each) and 80 hours on the project.	
Literature and Study Materials	The course will be based on research papers and web tutorials. An optional book is: Optical Wireless Communications: System and Channel Modelling with MATLAB® by Z. Ghassemlooy, W. Popoola, S. Rajbhandari	
Assessment	There will be two labs and one project. Each lab accounts for 20% of the grade (40% in total) and the final project accounts for the remaining 60%. The labs will consist of guided sessions (4 hours each) and independent work (18 hours). For the guided sessions, we will provide instructions, the HW platforms and a basic SW structure. Lab 1: 20% of grade Goal: Build a basic LED-to-PD link Timeline: Starts after Lecture 2 and is due after Lecture 4 Deliverables: Demo, short report, and Q&A Lab 2: 20% of grade Goal: Build a basic LED-to-Camera link Timeline: Starts after Lecture 5 and is due after Lecture 7 Deliverables: Demo, short report, and Q&A Final Project: 60% of grade Open. You will build an application on top of one of the two labs. Timeline: Starts after the second Lab and is due before the exam week. Deliverables: Demo (30%), Report (15%) & Oral exam (15%)	
Enrolment / Application	1. You need to enrol in Brightspace 2. The first lecture will be compulsory 3. This course can only accommodate 30 students, with ES students having a preference when demand exceeds capacity.	

IMPORTANT: The study guide information may change depending on the developments around the coronavirus.

ET4285	Measuring and Simulating the Internet	4
Responsible Instructor	Prof.dr.ir. F.A. Kuipers	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	(Advanced) Networking course (e.g., CS4055) and Programming skills.	
Course Contents	The Internet is a complex network without a fixed structure. Hence, measuring the Internet is crucial to acquire knowledge about the Internet infrastructure (topology), traffic, and performance (e.g., loss, delay, bandwidth, etc.). This course will discuss the design requirements and challenges in measuring and simulating the Internet, and the existing measurement methodologies (how/where/when to measure). Knowledge of how to conduct and evaluate Internet measurements enables the design and enhancement of a large set of applications, including: capacity planning and traffic engineering, network management and trouble-shooting, detecting network abuse and intrusions, etc.	
Study Goals	The goal of this course is to introduce the students to basic Internet measurement tools, as well as the state-of-the-art in Internet measurements research. The students will learn several Internet measurement techniques (e.g., active vs. passive measurements), and different software tools. Through a measurement assignment, the students will learn how to define/formulate a research problem, choose a specific approach, and complete a measurements-related research project.	
Education Method	Weekly instructions (8x2 hours) + independent project work (8x12 hours).	
Literature and Study Materials	Papers	
Assessment	Groups of students will be assigned a project that requires the students to put the theory on measuring and simulating the Internet into practice. The students have approximately 1 month to complete their assignment. The final assessment is based on the presentation (via report and/or demonstration) of the project assignment results and on the individual contribution and level of participation. Students within a group may thus receive different grades. As this is a project-based course, there is no official resit scheduled. Instead, an opportunity will be given to improve the work.	
Remarks	Disclaimer: The information about ET4285 may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	Because this is a project-based course, we can only admit a limited number of students (typically around 30, but the actual number depends on the number of TAs involved). If more students enrol, we will give preference to those who have successfully completed CS4055.	

ET4388	Ad-hoc Networks	5
Responsible Instructor	Dr. R.R. Venkatesha Prasad	
Contact Hours / Week x/x/x/x	3/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Wireless communications and networking Computer communication principles, Layering principle of Computer Networks. Digital communication.	
Course Contents	IMPORTANT NOTICE ------------------------- Please note that the prevailing conditions of the COVID19 may force us to modify the study guide information. It may change depending on the developments around the coronavirus. We may have to make changes to the teaching methodology, assessment, practical work, assignment and group activities. This will be instructed by the faculty/university management based on the orders of the government from time to time to protect the faculty and students. The above applies to all the fields in this coursebase for this course. The course will be offered ONLINE. Face to Face meeting may be possible depending on the situation [Safety of everyone is the highest priority]. -- Ad-hoc networks are formed in situations where mobile computing devices require networking applications when a fixed network infrastructure is not available or not preferred to be used. In such cases, mobile devices may possibly set up an ad hoc network themselves. Ad-hoc networks are decentralized, self-organizing networks and are capable of forming a communication network without relying on any fixed infrastructure. Ad-hoc networks form a relatively new field of research. In this lecture, besides general introduction to ad-hoc networks and their applications, we will focus on state-of-the-art methods and technologies for forming an ad-hoc network and maintaining its stability despite the dynamics of the network. The contents of the course are as follows: Positioning and applications (Chapter 1, 2 & 3 of the textbook, these topics are basics & pre-requisites; And Chapter 5) o Definition of ad-hoc networks o Comparison with infrastructure based systems o Typical applications o Advantages and challenges o Radio technologies for ad-hoc networks o Wi-Fi, Zigbee, Bluetooth Modelling ad-hoc networks o Propagation models o Topology models based on graph theory o Degree and hopcount o Connectivity theorems MAC protocols for ad-hoc networks (Chapter 6, 10 of the textbook) o Introduction to MAC protocols o Issues and design goals o Classification o Directional, muli-channel MAC protocols o Energy efficiency in MAC protocols o Quality of service Self organisation and Routing (Chapter 7, 8, 11 of the textbook) o Flooding o Node discovery, neighbour discovery o Route establishment o Topology maintenance, localisation o Proactive, reactive and hybrid routing o Typical protocols o Energy efficiency in routing o Broadcast and multicast o Effects of mobility on connectivity and capacity o Effect of nodes joining and leaving the network Advanced issues in ad hoc networks o Wireless sensor networks (Chapter 12 of the textbook and papers) o Cooperation (Reference papers) o Simulating ad hoc networks as part of project (optional: ns3, OMNET, OPNET) o Energy Harvesting Project presentations by students	
Study Goals	By the end of this course students should be able to: - Model the ad-hoc networks using Graphs. - Describe the working principles of medium access control protocols for ad-hoc networks - Explain the working principles, advantages and disadvantages of different classes of routing protocols for ad-hoc networks - Choose various components to form a coherent ad hoc networking architecture - Develop a simulator to evaluate the MAC and routing protocols for ad hoc networks - Assess the suitability of ad-hoc networks for different communication needs and scenarios	
Education Method	The course will be taught in lecture form. The presence of students at all lectures is required for optimum result. Students are required to participate actively in various forms of activities and peer-learning. New forms of teaching aids are used.	
Literature and Study Materials	1. Textbook: Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004.	

	2. Lecture notes consisting of slides presented at the lectures (Slides are only teaching aid and they are not substitute for textbooks, research papers, etc). 3. Some recent journal papers 4. Optional Reference Books 4.1. Distributed Algorithms, Nancy A. Lynch, Morgan Kaufmann, 1996 (for networking algorithms) 4.2. Ad Hoc Mobile Wireless Networks, Principles, Protocols and Applications by Subir Kumar Sarkar , C Puttamadappa , and T. G Basavaraju, Auerbach Publications, 2008. This book is avaliable online in the library. 4.3. Wireless Ad Hoc and Sensor Networks, A Cross-Layer Design Perspective by Jurdak, Raja, Springer, 2007. This book is avaliable online in the library. 4.4. Ad-hoc Networks: Fundamental Properties and Network Topologies, by Ramin Hekmat, Springer. 5. OPNET/ns-2 web pages, tutorials and video lectures
Books	Ad Hoc Wireless Networks, Architectures and Protocols by C. Siva Ram Murthy and B.S.Manoj, Prentice-Hall Pearson, 2004. However, I also use other materials from Internet and other books listed above.
Assessment	1. There will be written tests/examinations for this course. 2. The students will carry out a project in a group and submit a short report. 3. Participation in off-track discussions on Facebook/Brightspace/FeedbackFruits and wikis. Final score is based on marks obtained during tests, project, assignment (in groups) and bonus marks. All the details will be given in the first class. Breakup: 2 Tests + Final Exam = 55% Project 40% Self assessment + Reflection 3% Activities on Feedback Fruits or any Online platform 2% ====================== Changes due to COVID19 in 2021 ================ There are three chapters Chapter 1: Network modelling Chapter 2: MAC protocol Chapter 3: Routing + Lab: Simple experiments using laptops (individual) + simulations (group-wise) Marks Breakup -------------- Homework/Assignments/Group works Part A1: Group-wise assignment. 3 times (1 per chapter) -- 15 marks Part A2: Group-wise Q&A. 3 times (1 per chapter) -- 30 marks Part B1: Individual experiment+report Part B2: Group-wise simulations + report + demo Part B1 + Part B2 - 50 Marks Part C: Self-assessment + peer activities -- 5 marks Resit: Part A1 & A2 will not be repeated. Only Part B1 & B2 are allowed for the Resit this year, because of COVID19; and the projects should be done individually in Resit. == (More information will be given in the first class) disclaimer: Information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	Different conditions for different test/exams. Conditions will be informed 1 week before the exams/test.

ET4394	Wireless IoT and Local Area Networks	5
Responsible Instructor	Dr. P. Pawelczak	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Students are advised to follow the course Wireless Communications (ET4358) before taking this Wireless Networking course. An advantage is to have entry-level programming skills (Matlab, Python, C/C++). Nonetheless, students with little knowledge of programming will be helped.	
Course Contents	DISCLAIMER: this study guide information may change depending on the developments around the corona virus. The following modules will be discussed during the lectures: Introduction (example topics): - What is wireless networking - Where to search for (academic) wireless network literature and resources Medium Access Control (example topics): - WiFi: hidden/exposed terminal problem, Carrier Sense Multiple Access - Bluetooth standard: in-depth look into the channel hopping, protocol specifications WiFi (example topics): - Review of IEEE 802.11 standards - Protocol format - ISM band regulation - Adaptive Modulation and Coding - WiFi Matlab class (assignment) IoT networking standards (example topics): - LoRa: protocol specifications, energy consumption, modulation format, network design Review of wireless tools (example topics): - Introduction to wireless packet sniffing and analysis using Wireshark (assignment) - Simple simulations of WiFi network with NS3 RFID networking (example topics): - Principles of backscatter - Protocol formats: EPC C1G2 - RFID hackathon (assignment) Cognitive radio (example topics): - Basics of spectrum management - White Space Databases - Theory of spectrum sensing	
Study Goals	At the end of the course students will be able to: (i) to understand how practical wireless systems work and get a deeper understanding of how the theoretical concepts of wireless communications apply to practice; (ii) employ their own analysis methodology to assess new wireless network systems (especially at the physical layer); (iii) understand rapid prototyping of new wireless systems (for instance, with software defined radio).	
Education Method	Lecture presentations, mini-project assignments, assigned paper reading and its critical analysis and presentation.	
Computer Use	Each student should have its own laptop (preferably with a Linux distribution, where Linux must not be installed on a virtual machine). We will be using Matlab, and/or NS3 and/or GNURadio and/or Wireshark for the assignments.	
Books	WiFi Matlab WLAN toolbox: https://nl.mathworks.com/help/wlan/; Wireshark learn page: https://www.wireshark.org/#learnWS; tutorial on NS3 network simulator: https://www.nsnam.org/documentation/; specific chapters from books provided at the beginning of each lecture.	
Prerequisites	Background in programming (Matlab, Python, Bash)	
Assessment	Points from the mini-project assignments. A research paper analysis from conferences such as IEEE INFOCOM, ACM MobiCom, ACM SIGCOMM will be required to pass the course. disclaimer: information may change depending on the developments around the coronavirus.	

IN4254	Smart Phone Sensing	5
Responsible Instructor	M.A. Zuñiga Zamalloa	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Requirement 1: Students MUST either (1.1) have passed a JAVA programming course, or (1.2) have passed a C/C++ programming course and be familiar with JAVA, or (1.3) know Objective C (programming language for MACs). This requirement is equivalent to having passed the course TI 1206 in our first year Bachelor curriculum "Object Oriented Programming" Requirement 2: Students MUST (2.1) have passed a basic course on Probability Theory. This requirement is equivalent to having passed the course TI 2216M in our second year Bachelor curriculum "Probability and Statistics". We will be refreshing some concepts on Probability, but we will not be refreshing concepts on Object Oriented Programming.	
Course Contents	The course provides an introduction to the current research trends in the area of smartphones. The course will be based on a programming project, where students will form groups of two to develop a smartphone application. This is not a programming course; students are expected to have already programming experience. To develop a smartphone application, a user needs to be familiar with (1) the signals and data that smartphones can gather, and (2) the mathematical tools necessary to process this data. This course will provide a solid background for the above two points. During the lectures we will analyze the latest research papers on this emerging field. We will dissect these papers to understand how techniques from algorithms, signal processing and machine learning are used to develop some exciting applications. The students will then use these basic technical tools to develop their own apps.	
Study Goals	The goals of this course are twofold. First, to expose students to the increasingly important area of mobile computing. Students will learn how mobile phones can be used to solve problems in areas ranging from health care and indoor localization to song recognition and traffic management. Second, to provide students with a basic set of tools to develop their own applications. For students aiming for industry, the course should enhance their ability to use theoretical tools to solve practical problems. For students involved on research activities, the course will provide them with the necessary background to use smartphones as a distributed sensing and processing unit that could be used to solve the particular problems in their areas. After taking this course students will be able to: (1) Explain the current applications, methods and research trends in the area of smartphone sensing. (2) Apply key mathematical tools in the development of smartphone applications. (3) Analyze how a sensing and computing problem can be solved via the use of smartphones, and identify the steps required to design a solution. (4) Create a non--trivial and innovative smartphone application.	
Education Method	Lectures + Lab The project work, including the written report, covers the entire duration of the course period, and will take approximately 120 hours, of which 14 hours are spent on lectures, 10 hours preparing reports, 10 hours reading research papers, and the remaining part programming the App (the time spent in the Lab belong to this latter part).	
Literature and Study Materials	Research Papers and web tutorials	
Assessment	Written reports + project presentation + oral exam Overall, the final grade is determined by: 1) Two intermediate reports (5% of grade each, 2 pages each) 2) Final report (10 % of grade, 5 pages) 3) Final project demonstration (80% of grade) The first two reports are due on the third and fifth week; and the final report, project and exam are due on the ninth week. There is no resit for this course. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	1. You need to enrol in Brightspace 2. The first lecture will be compulsory 3. This course can only accommodate 60 students, with ES students having a preference when demand exceeds capacity. If your program marks this course as required, you are guaranteed a spot. IMPORTANT: The study guide information may change depending on the developments around the coronavirus.	

IN4343	Real-time Systems	5
Responsible Instructor	Dr. G. Iosifidis	
Instructor	Prof.dr. K.G. Langendoen	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures & 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	3TU MSc Embedded Systems; the corresponding courses are 2IN26 at TU Eindhoven, and 312030 at TU Twente	
Expected prior knowledge	Basic software engineering, C system programming, basic Linux operating system knowledge	
Course Contents	- basic concepts of RTS - worst case execution time estimation - scheduling policies - response-time analysis - jitter analysis - handling overload - multiprocessor scheduling - reservation-based scheduling	
Study Goals	The course intends to bring the student into the position to: - Explain the fundamental concepts and terminology of real-time systems - Construct task schedules using different scheduling policies under a given set of realistic system constraints - Analyze the timing behavior of a system for a given system model and scheduling policy - Discuss advantages and disadvantages of different scheduling policies for a given platform or system - Discuss the effect of hardware and software interferences on the timing behavior of a given system - Identify (reverse engineer) parameters of a scheduling scheme or a task set from output traces of the system - Derive (reverse engineer) the system specification from a given implementation (in the lab) - Evaluate the scheduling overheads of a given implementation (in the lab) - Implement event-based scheduling policies on a given microcontroller (in the lab)	
Education Method	lectures with exercises (32 hrs); self study (78 hrs); lab assignments (30 hrs)	
Books	Hard Real-Time Computing Systems by G.C. Buttazzo, Springer 2011	
Assessment	Written exam (grade) + lab work; the exam has a resit disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	3	
Permitted Materials during Tests	Simple calculator	

IN4390	Quantitative Evaluation of Embedded Systems	5
Responsible Instructor	Prof.dr. K.G. Langendoen	
Instructor	Dr. G. Iosifidis	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	For this course, a basic knowledge of linear algebra, probability theory and automata / transition system theory is assumed. However, there is no formal requirement on having followed specific courses on these subjects.	
Course Contents	This course introduces students to the formal methods and tools that can be used to assess the performance of embedded systems. The list of covered topics includes: - Design of Experiments - Petri Nets - Markov Chains - Queuing Theory	
Study Goals	At the end of the course, the student has a good overview over the kind of formalisms that are used when quantitative aspects like time, probability and resource usage play a role in the analysis of system behavior. The student knows how to use particular examples of such formalisms, and is aware of their limitations. Also, the student has gained experience with the use of several analysis tools for verification and validation of quantitative formal models.	
Education Method	Lectures + Hands-on Sessions - 14 lectures in 2 blocks of 2 hours per week - 5 lab sessions of 4 hours each - take home assignments/projects	
Literature and Study Materials	- reading list of papers and book recommendations	
Assessment	In-class quizzes (graded) A written exam (3 hours) A set of mandatory practicals (pass|fail) A set of customizable assignments (graded) disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	The exam is closed book, but a cheatsheet with relevant equations will be provided, and a simple calculator may be used.	
Enrolment / Application		

IN4398	Advanced Practical IoT and Seminar	5
Responsible Instructor	Dr. R.R. Venkatesha Prasad	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	fundamental understanding of wireless communications, familiarity with wireless communications and embedded systems and knowledge of Android programming/Python/C++/Matlab.	
Parts	Each seminar: 2x 45 minutes (2 parts) + 10 minute break	
Summary	This course is an involved hands on course for self motivated students. Students work in a group of two (max 3) usually. Students are expected to have sufficient programming and hardware development skills. The course is project oriented thus students are expected to deliver a working model with demonstrations. Preference will be given to Q5 students. Q1 students are discouraged unless otherwise explicitly allowed. -- IMPORTANT NOTICE REGARDING ADJUSTMENTS DURING THIS COVID19 SITUATION The course will be offered ONLINE. Depending on the COVID19 situation there may be on-campus meetings/discussions [Safety of everyone is the highest priority]. -- This course is done in a group of two (max 3). Instructor/TA will help connecting a potential groupmate in the first class using Brightspace/Feedback Fruits. Activities: How to in Groups? ----------------------------- Part 1: Seminar We guide the groups to select a paper. Students need to prepare slides and deliver a seminar using online tools; it is a group activity. For each seminar all the students MUST be present. Absenting without permission will be taken seriously. Part 2: Project (1). Students can share the responsibilities if they can organize in such a way that one works on H/w and another on software so that they develop the project together. Meeting each other may be possible. However such face-to-face meetings should be done by strictly adhering to the COVID19 related instructions from the university/government. (2). Students may exchange the components/boards within the group but carefully adhering to 1.5m rule. The decision to do such cooperation is left to the judgement of the students and it is their own responsibility. Please note safety is first. For Students not in NL or not able to be in Delft: (3). Instructor/TA would be offering an opportunity to do a hardware project if students can buy simple hardware like Arduino boards/raspberry pi, etc. They should try to use the method as in (1) above. (4). Instructor/TA may offer Algorithm/simulations/android programmes/Contiki based networks or Open testbeds projects. In these cases, online group activities are possible. NOTE: In some extreme cases, Instructor/TA may allow one person project after assessing the situation and after discussing with the student. Supply of Hardware: ------------------- 1. Instructor/TA is able to provide take home components (limited numbers) like Arduino boards and some sensors. We may provide some other instruments/sensors/boards depending on the selected project. These could be collected from TA while adhering to sanitization and social distancing rules. 2. Students may also use their own components.	
Course Contents	Course will be composed of a series of seminars related to the broad topic of the Internet of Things. Students will present their results on investigations regarding the possible extension of the ideas presented in the assigned papers.	
Study Goals	To be able to design components of Internet of Things and showcase an application or product through an implementation of a project. Specifically, to be able to bring entrepreneurial aspect of the project and also to be able to evaluate the project in depth. To be able to criticize and assess system-level components of the Internet of Things environment discussed in the scientific literature.	
Education Method	Seminar will be composed of (i) seminar presentation on a selected research paper (from top journals/conferences) presented individually by students and, (ii) work on a research project. Students will be provided with a list of projects that will be assigned to them. Project will be summarized in the form of a written report (report must include critical analysis). Within a project any hardware/software platform can be used and demonstrated. User experience/study, where applicable, also needs to be executed. Selected paper needs to be critically evaluated and a proposal to extend the assigned paper will need to be presented in a form of a presentation. Paper extension should focus on a system level idea. Presentation skills, thinking and reflection abilities are looked into carefully. The teams are composed of two (or three) students generally. NOTE: The total amount of work would be on the higher side of 150 hours; since this is an advanced course, students are expected to already have very good knowledge of hardware platforms, coding, and design environment. In case of lacking in some of these skills, we expect students to acquire them outside this budgeted 150Hrs. However, the number of hours of workload mentioned here is ONLY a guide, the efforts depend on the project, goals, and the collaboration with the team member(s).	
Literature and Study Materials	This is a project based course. Thus, Internet and other appropriate manuals (for chipsets, etc.) would be useful. For papers to read and present, we expect students to look into: Infocom, Mobicom, IPSN, Sensys, Sensapp, Mobihoc conference papers. Journals: IEEE Trans on Networking, Trans on Mobile computing, JSAC, etc. ACM Trans on CPS, etc.	
Assessment	Part 1: Assessment based on presentation quality, slides, Q/A, constructive criticisms, ideas for improvements, etc.	

	Part 2: Project execution in a group, demonstration, Q/A and a report describing the outcome of the assigned project. Part 1: 30% of the whole marks; Part 2: 60% of the whole marks. In the assessment, a focus on the practicality and entrepreneurial aspect of the idea will be prevailing. A working model, demonstration, and a report, are expected. Individual Q/A after demonstrations will be part of the assessment. Part 3: Up to 10% marks would be awarded if the report is detailed above the minimum expected and the work is ready for a submission to a conference. Report will not have individual component, however, Q/A may have individual component. Please NOTE: There would be no resit since this is a hands-on course being executed in a team. disclaimer: Information may change depending on the developments around the coronavirus.
Tags	Circuits Group work Programming Project Software
Judgement	30 Marks for Seminar 60 Marks for project, demonstration, Q/A 10 Marks for making a paper that could be submitted as a manuscript for review to a conference/journal. Bonus marks are offered for excellence in various components during the course. That would add to the total marks not exceeding a max of 100.
maximum aantal deelnemers	Max 30 students, which translates to 10-15 groups (Groups consisting of 2 or 3 students).

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Interactive Intelligence 2021	
Introduction 1	CANCELLED: CS4170 Seminar Intimate Computing in 2018-2019. Please note that due to the unforeseen circumstances course CS4170 Seminar Intimate Computing (Q4) will NOT take place this academic year 2018-2019. Therefore this course is not listed.

CS4015	Behaviour Change Support Systems	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	none	
Course Contents	Behavior change support systems (BCSS) are computer-based systems that support individuals to form, alter or reinforce cognitions, attitudes or behaviors without using coercion or deception. They can serve individuals throughout the various stages of a change process, such as awareness developing, contemplation, action strategy development, development of new behaviors, and maintaining these new behaviors. Virtual healthcare coaches, negotiation support systems, and applications that provide individuals with personalize financial guidance are three examples of these systems. To establish, modify or maintain change BCSS can deploy computerized persuasive strategies (e.g. reducing effort to establish target behavior, or argumentation and reflection strategies), simulations (e.g. serious gaming, virtual reality), relational software agents (e.g. ePartners, virtual coaches), and personalization based on longitudinal user data. BCSS are found in many domains, including education, sales, negotiation, management, and particular in the health domain.	
Study Goals	The course allows students to achieve understanding of principles, concepts and theories underlying BCCS systems and methods for designing them.	
Education Method	In the pre-recorded video material, theories, principles and methods are presented, discussed and illustrated with examples from the field. The video material is support by online self-tests. In the lectures, teacher and students discuss and make assignment related to pre-recorded material of that week. At home students work on their own in small groups on coursework assignments to develop a product design for a BCSS. In the practicum session student groups presented the progress on their coursework and receive feedback. Expected Workload Pre-recorded video material: 18 hours (2 hours × 9 lectures) lecture: 10 hours (10 × 1 hours) practicum 14 hours (7 × 2 hours) Reading time: 36 hours (9 × 4 hours reading time) Preparation presentation: 3 hours (3 × 1 hours for each presentation) Coursework project, including writing report, and final presentation: 50 hours Exam preparation and revision: 9 hours Total = 140 hours	
Literature and Study Materials	Will be announced on brightspace	
Books	Wendel, S. (2013). Designing for Behavior Change: Applying Psychology and Behavioral Economics. " O'Reilly Media, Inc.".	
Assessment	The course is assessed by coursework and an exam as follows: (60%) computerised examination (or oral exam) (40%) Coursework Project (resulting in a report, and final presentation include question and answer round where individual group members are assessed on coursework) If the expected number of students registering for exam or resit is small, the teacher might decide to replace the computerized examination by an oral examination. Separate marks will be given for exam and coursework, only a combined mark is recorded in Osiris. A passing final grade for the course can only be earned when for both the exam and coursework at least a 5.0 is earned, and the weighted grade for exam and coursework is at least a 5.8. Resit coursework A second submission of modified coursework is only allowed for coursework that received a fail mark (<5.8) for the first submission. Overall resit mark for coursework will be capped to 5.8. Note that individual marks for coursework or computerised exam (or oral exam) do not carry to the next year. disclaimer: information may change depending on the developments around the coronavirus.	
Exam Hours	2	
Co-Instructor	M.L. Tielman	

CS4125	Seminar Research Methodology for Data Science	5
Responsible Instructor	Dr.ir. W.P. Brinkman	
Instructor	Dr. K.A. Hildebrandt	
Instructor	J. Urbano Merino	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	basic knowledge in mathematics (linear algebra, calculus, probability and statistics)	
Course Contents	The course focuses on research methods for data science. It looks at underlying principles and concepts for data collection, analysis and data processing, as well as the use of tools to do this. The main topics of study are: Conceptualizing research questions and experimental design Frequentist and Bayesian data analysis Generalized linear models for statistical analysis Multilevel modelling for hierarchical and longitudinal data analysis Measuring and sampling, validity and reliability Linear and nonlinear dimensional reduction Principles of statistical testing In the course, students will be using software tools such as R, and Matlab/Mathematica	
Study Goals	The main aims of this module for the student is to achieve understanding of research methods for data science and obtain practical experience with data analysis and data processing methods. This module provides students with the opportunity to develop and demonstrate their understanding, knowledge, and competence. The learning outcomes for the module are that students will be able to: 1. Appreciate and comprehend strategies for collecting and processing data to answer data-driven research questions 2. Understand and reproduce key principles underlying statistical data and data processing analysis 3. Learn to identify and avoid typical biases, paradoxes and misunderstandings in data-driven research 4. Apply and select appropriate data modelling techniques to analyse data and data processing	
Education Method	Lectures/Assignments Expected Workload Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours (5 × 5 hours for each tool) Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours) Total = 140 hours	
Literature and Study Materials	Will be provided online	
Assessment	Course will be assessed on 3 coursework assignments. A) Analysis of experimental research data (40%) B) Exploration of real-world data set (20%) C) Linear and nonlinear dimensional reduction (40%) Students work in small groups on the 3 assignments. For each assignment, the student group submit a report and give a presentation including a question and answer round where individual group members are assessed on the coursework. The final course mark is the weighted average of the three assignment marks. Note that, there is a minimum grade of 5.0 for each assignment grade and an average grade for all components of at least a 5.8 in order to pass the course. Also, marks for individual assignments do not carry to the next year. Resit next quarter Resubmission of modified coursework is only allowed for assignments that received a fail mark (<5.0). Overall resit mark will be capped to 6.0. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	NA	

CS4165	Seminar Social Signal Processing	5
Responsible Instructor	Dr. H.S. Hung	
Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Contact Hours / Week x/x/x/x	2/2/0/0 + project	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Your background should consist of a combination of at least two of these topics or related topics: Signal Processing, Speech/Audio Processing, Computer Vision, AI, Machine Learning, Pattern Recognition, Reinforcement Learning, Deep Learning/ Neural Networks, Cognitive Modelling. These can be topics that you learned about at either Bachelor or Master level.	
Course Contents	The core of social intelligence is our ability to understand and interpret social signals of a person we are communicating with is. Social intelligence is a facet of human intelligence that has been argued to be indispensable and perhaps the most important for success in life. Social Signal Processing (SSP), the new, emerging, domain aimed at understanding social interactions through machine analysis and production of nonverbal behavior. In this course you will learn how next-generation computing can make use of such social signals by giving it the ability to recognize and produce human social signals and social behaviors. Think about turn taking, politeness, disagreement, emotions, rapport. You will learn about relevant findings in social psychology, and you will learn computational techniques that allow systems to make use of social signals to become more effective and more efficient by being able to detect but also simulate (e.g. in virtual agents) blinks, smiles, crossed arms, laughter. Socially aware computing. These techniques can be used in robots, virtual agents, smart homes, crowd monitoring, etc.	
Study Goals	Know what social signals are. Be able to apply computational methods to detect and simulate such signals. Position the field of social signal processing in computer science and psychology, and identify its major goals and angles of study. Define and explain social signals in humans and know about major psychological theories of social interaction. Explain major social signal recognition, simulation and expression techniques in computational systems. Develop (in groups) a research project that uses social signals in a non-trivial manner with hypotheses, research questions, and a supporting literature survey, and together evaluate the resulting system. These are important skills that prepare students towards their own masters thesis study later on.	
Education Method	This course is run as is a two quarter course running in Q1 and Q2. The course has been historically open to students from Leiden University as we see that mixed university groups leads to better quality projects and peer learning, which is a key part of the course. The course has therefore been designed to block off Monday afternoons where lecture times are scheduled to allow travel between the two institutions as part of the timetabled course. In light of the move to have campus education again, the lectures will be on campus and not virtual. The course has historically been run also as a Q1 only course. However, we have found that the 2 quarter model allows students more time to learn and absorb the course learning objectives leading to higher quality projects. In practice most of the contact hours are in Q1 with more time being devoted to the project work in Q2 with occasional progress meetings. Seminar: 2 hours of lectures per week for most of Q1. Self-study of papers. The papers will be made available at the start of each lecture. Project: 2 hours of class contact hours every 2-3 weeks for latter part of Q1 and Q2. Perform a piece of research (survey, research question, programming, testing) and write a paper about it. Students will work in teams of about 3-4 persons! Depending on the total number of students enrolled, teams will either work on a topic of their own or we will all together work on one big topic.	
Literature and Study Materials	Selected papers made available before the course.	
Assessment	UPDATED as a result of on campus education. 10% mini exam. This multiple choice assessment helps to establish basic knowledge of the course material before the project work starts. 40% project proposal quality (mid term presentation + survey) 50% Project execution. No final exam. disclaimer: information may change depending on the developments around the coronavirus.	
Maximum number of participants	60	

CS4235	Socio-Cognitive Engineering	5
Responsible Instructor	Prof.dr. M.A. Neerincx	
Instructor	Dr.ir. W.P. Brinkman	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Basic prior knowledge on human-computer interaction is helpful, but not required.	
Course Contents	Whether you are playing a game in virtual reality, driving a semi-autonomous car, educating yourself in a MOOC, or harmonizing your health and lifestyle via a mobile app; nowadays intelligent networked information and communication technology is omnipresent. This course focuses on the design of human-aware intelligence into such environments, to support joint human-technology performances that bring about positive human experiences (such as social robots that help activity coordinators guide health-promoting games for people with dementia, http://rejam.tudelft.nl). In the Socio-Cognitive Engineering (SCE) course (MSc level), you will become acquainted with the application of a coherent set of methods for the design and evaluation of human--agent collaboration. Based on the SCE-method, we will elaborate on the state of the art of intelligent user interfaces (ePartners), such as artificial personal assistants, artificial team mates, eCoaches, social intelligence, and companion agents. The main topics of study are: - Design methods: Cognitive Engineering, Value Sensitive Design, Scenario-based Design, Claims Analysis, Design Rationale, Design Patterns. - Design for collective intelligence: Knowledge Representation, Ontology Engineering, Mental Models, Theory of Mind, ePartners, Adaptive Automation, Socially Intelligent Interfaces. - Design Evaluation: Prototyping, Test Methods, Measures, Questionnaires, Ethics. - Human Factors Theories and Models: Human Cognition & Learning, Memory, Emotion, Task Load, Human-Agent Teamwork, Behavior Change and Persuasive Technology.	
Study Goals	At the end of this course, students will be able to: 1. Explain the essential concepts of the design methods addressed in the course. 2. Explain the (dis)advantages of various design methods and their complementarity. 3. Apply the design methods addressed in the course in their research and design projects. 4. Explain what a design rationale is. 5. Construct a design rationale. 6. Create design specifications that are grounded in a design rationale. 7. Evaluate the strengths and weaknesses of a design rationale, e.g. using human-centered evaluations that test the design rationale. 8. Explain some of the state of the art human factors theories, models, and methods relevant to intelligent user interfaces, human-agent collaboration, and ePartner technology. 9. Write a structured report about a design-test cycle, with sufficient detail for a new group of researchers to continue the research. 10. Present work on a design project to an academic audience. 11. Work in a group on collaborative assignments.	
Education Method	LECTURES During the lectures, the teachers will present a range of theories, models, and methods relevant to socio-cognitive engineering. Students are required to read a number of scientific papers which are made available on Brightspace, along with the sheets/slides of the lectures. Together, the sheets/slides and the papers provide the students with the required theoretical knowledge to work on the practical project, and to learn about relevant design methods, human factors theories, conceptual solutions, and design principles. Most of the lectures include practical assignments and discussions stimulating the students to apply the contents of the lecture to their project (also see Project). PROJECT In the project, students work in groups to apply the knowledge acquired during the lectures. Students are required to plan, execute, present, and report on a complete design cycle (i.e. design, prototype, and evaluation) for a given design problem. This year (like the past years), the design problem is a social robot for older adults with dementia, and their social environment (https://rejam.tudelft.nl). The objective of the social robot is to improve humans physical, social, cognitive, and emotional well-being. The students will use the Wiki Socio-Cognitive Engineering (WiSCE) tool to specify the design rationale and its evaluation, step-by-step (see also https://scetool.ewi.tudelft.nl). Throughout the course, students will give presentations about their progress, on the design and evaluation of their prototype.	
Literature and Study Materials	Papers from scientific journals on Brightspace. Lecture notes on Brightspace.	
Assessment	Literature and study material consist of: - Papers from scientific journals on Brightspace. - Lecture notes on Brightspace The module assessment concerns the processing and application of the theory and methods; the construction of the design (rationale) and the evaluation; and the provision of the resulting concise and coherent report (including the lessons learned): Presentations (10%) Prototype (10%) Project report according to the prescribed format (70%) Individual reflection (10%)	
Exam Hours	There is no exam. The assessment is based on a paper, presentation and report. During the course, students will receive feedback on interim work. There is no resit after the end of the course.	

CS4270	Conversational Agents	5
Responsible Instructor	Dr. C.R.M.M. Oertel Genannt Bierbach	
Instructor	Dr. M. Bruijnes	
Instructor	Dr. P.K. Murukannaiah	
Instructor	M.L. Tielman	
Co-responsible for assignments	F. Broz	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic programming skills (e.g. Python and Java) Probability theory and statistics	
Course Contents	Chatbots, embodied and conversational virtual agents, and social robots are becoming more and more popular. Many people are owning an Alexa, Cortana or Echo or are talking to their virtual assistant on their phone. Indeed, such technologies have the potential of making our lives easier and relieve people from the more repetitive tasks. For example, it is imaginable that such systems are being used for financial applications by helping customers with frequently asked questions but also to advise them on in the long term more impactful decisions such as their pension plans. Further applications can be imagined in the area of healthcare and education, some of which are already in existence today. In this course, attention will be given to different verbal and nonverbal behavioral characteristics, like speech, intonation, gaze and gestures that humans show when communicating with both other people and machines. This behavior is then related to different dialogue functions, including turn-taking, addressing others, and backchanneling, that give shape to the communication process. This course introduces conversational agent technology. We cover agent related technologies which can be grouped into: Dialog Management NLP speech synthesis social robotics	
Study Goals	After this course you have learned to: 1) Apply relevant linguistic and psychological theory to conversational agent systems 2) Analyse human-human conversational data to better design ML models 3) Explain which components are part of a dialog system and what distinguishes rule-based from statistical dialog systems 4) Describe the design and implementation of state-of-the-art conversational agents, give examples of their application areas and analyse and discuss the limitations of current systems 5) Evaluate the effects of affect and embodiment on human-agent interaction 6) Create and evaluate a socially-aware conversational agent by applying state of the art tools and libraries	
Education Method	There are 2 lectures and 1 practicum scheduled per week. Students work in groups of 3-4 on a group project. Lectures: 26 hours (13 × 2 hours lectures) Reading time: 39 hours Preparation basis tool use: 25 hours Coursework project, including writing report and prepare for presentation: 50 hours (10 × 5 hours)	
Literature and Study Materials	We use the book "The conversational interface " by Michael McTear, Zoraida Callejas, David Griol. This book is freely available through the TU Delft library. https://link-springer-com.tudelft.idm.oclc.org/book/10.1007%2F978-3-319-32967-3 Other relevant material will be provided on Brightspace.	
Assessment	Online Examination (30%) Group Assignment (50%) (This assignment will result in a group report and a group online demonstration with individual question/answer part) Group presentation (20%) The exam and the assignments are both intermediate results, and will not count separately for the next academic year. Only the final grade will remain. A passing final grade for this course can only be earned when for the online examination and the group assignment at least a 5,0 is earned, and the average grade for both is at least a 5,8. Projectwork with a mark lower than 5.8 can be modified and resubmitted. The mark for resubmitted coursework will be capped to 5.8 Note that individual marks for projectwork or written exam do not carry to the next year. We further grade the labs as pass/fail. By a successful pass of all labs a bonus of 0.3 will be awarded towards the group assignment grade. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	Dr.ir. W.P. Brinkman	

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Multimedia Computing 2021

CS4065	Multimedia Search and Recommendation	5
Responsible Instructor	Prof.dr. A. Hanjalic	
Responsible Instructor	Dr.ir. C.C.S. Liem	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	Nowadays, a huge amount of multimedia data is available online. While this has the potential to serve a multitude of use cases, the sheer amount and diversity of available multimedia data and consumer information needs require the development of sophisticated access mechanisms. Furthermore, the term "multimedia" implies that user queries and data to be handled are rich and multimodal (combining text, image, video, audio, etc). In this course, methods, algorithms and best practices are discussed which deploy this richness of information to maximize the effectiveness, efficiency and intuitiveness of multimedia search and recommendation. Furthermore, implications of the fact that the data is consumed in networked communities of human users are treated. The course will both consider data analytics aspects for multimedia search and recommendation (with focus on emerging topics), as well as system and implementation aspects for multimedia search and recommendation (with focus on handling real-world multimedia data).	
Study Goals	Students will be able to explain the concept of multimedia; explain the principles underlying basic multimedia search engines; explain the functioning of basic multimedia recommender systems; describe and implement common representations of multimedia content; describe and implement common ranking mechanisms for multimedia search; describe and implement common recommender system techniques; describe and implement common social media analytics techniques for multimedia search and recommendation; interpret current academic literature in the field of multimedia search and recommendation; identify strengths and weaknesses of state-of-the-art multimedia search and recommendation functionalities; identify challenges belonging to the development of multimedia search and recommendation functionalities; identify evaluation criteria for multimedia search engines and recommender systems; explain the difference between topical relevance and utility in multimedia search and recommendation; describe and implement cross-disciplinary approaches to multimedia search and recommendation; describe and implement practical solutions to deal with real-world multimedia search and/or recommendation; propose and justify a vision on near-future improvement opportunities for a selected state-of-the-art multimedia search and/or recommendation analytics technique; develop a practical implementation based on an academic description of a selected state-of-the-art multimedia search and/or recommendation technique and assess it against a baseline on a real-world dataset.	
Education Method	lectures, lab course, final assignment	
Literature and Study Materials	Will be handed out by lecturers during the course	
Assessment	Group project, encompassing: an implementation of selected MMSR techniques on real-world data; a research proposal on an emerging topic in MMSR. In principle, a group grade will be given to the corresponding work, unless the teaching staff sees clear motivations for differentiation in grading. Lab assignments: pass/fail. Partial results towards a final course grade do not carry over to subsequent academic years. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.	
Judgement	Group project.	

CS4195	Modeling and Data Analysis in Complex Networks	5
Responsible Instructor	H. Wang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The assignment and final project require basic programming skill.	
Course Contents	Big Data is mostly obtained from features of components and the interactions among components in large complex systems. Examples are (1) end user features and interactions in both online and real-world social networks like Twitter, LinkedIn (2) data from content sharing platforms such as YouTube (3) physiological data of the brain and (4) stock prices etc. in economic systems. Such a dataset is networked in nature i.e. the data of the system components or interactions are (cor)related to each other. This course introduces the basic methodologies to analyze, model, interpret and possibly to predict such Networked Data, combining advances from network science, modeling of dynamic processes and statistical physics, beyond machine learning algorithms. These methods will be applied to diverse real-world datasets obtained from e.g. Facebook, LinkedIn, YouTube, the brain etc.	
Study Goals	After this course, students could construct a network based on the dataset, characterize and model the network in order to e.g. detect patterns and anomalies, model the data via dynamic processes (e.g. viral spreading) on networks to decode the underlying governing mechanisms of e.g. information/error/behavior contagion and to predict e.g. the popularity of a product, news, disease, computer virus, control the contagion process such as maximize the information prevalence and market share. Students could obtain an overview of the Msc/Phd projects on the frontiers of networked data analysis.	
Education Method	In total, there will be about 7 lectures. Students will also learn via an assignment and a final project (each group gets individual supervision).	
Assessment	The final grade is based on the assignment (20%) and final project (80%). There is no resit for both the assignment and the project. disclaimer: information may change depending on the developments around the coronavirus.	

IN4314	Seminar Selected Topics in Multimedia Computing	5
Responsible Instructor	P.S. Cesar Garcia	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	signal (image, audio) processing, pattern recognition, networking and distributed systems	
Course Contents	Through all the exciting recent advances in digital media technology and the rapid growth of social media platforms, multimedia content is increasingly embedded in our daily lives, gaining enormous potential in improving the traditional educational, professional, business, communication and entertainment processes. To be able to use this potential for transferring these processes into user-centric interactive multimedia applications, technology is required that can help us access, deliver, enrich and share rich-media content. This course provides insight into the state-of-the-art cross-disciplinary research efforts related to the development of such technology. The topics covered by the course include, but are not limited to, multimedia systems (transport and delivery, telepresence and VR, mobile), multimedia experiences (Quality of Experience, Collaboration), and multimedia engagement (emotional and social signals and social multimedia).	
Study Goals	To become acquainted with the state-of-the-art research and development activities in the field of Multimedia Computing, and to become an expert in one particular "hot topic", such that they are able to identify the "knowledge gap" (i.e., the place in which more research is needed in order to advance the state of the art).	
Education Method	readings, seminar discussions, presentations, survey paper	
Literature and Study Materials	Readings, possibly including video lectures.	
Assessment	The students demonstrate the knowledge that they have acquired by making a presentation on a pre-existing survey (10%), then writing their own survey on a new topic (65%), and finally by making a presentation on that topic (25%). The students must complete all three components. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Pattern Recognition & Bioinformatics 2021

CS4070	Multivariate Data Analysis	5
Responsible Instructor	Dr.ir. F.H. van der Meulen	
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	4/4/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	* Introduction Probability Theory and Statistics: see for instance A Modern Introduction to Probability and Statistics Understanding Why and How Series: Springer Texts in Statistics Dekking, F.M., Kraaikamp, C., Lopuhaä, H.P., Meester, L.E. 2005, XVI, 488 p. 120 illus., Hardcover ISBN: 1-85233-896-2 * Basic calculus * Linear Algebra: matrix multiplication, the inverse of a matrix, the transpose of a matrix, least square solution. see: David C. Lay: Linear Algebra and Its Applications ISBN-10: 0321385179 ISBN-13: 9780321385178 ©2012 Pearson)	
Course Contents	PART I: Stochastic models will be developed on the basis of probability theory. Probability theory describes the behavior of certain phenomena in terms of how likely it is that certain values will occur. Central features of the models will be discussed are random variables, probability density functions, and the expected value operator. In describing random processes and signals, the correlation function and conditional probabilities play a central role. It addresses the following subjects: 1. Random variables. Matlab exercise on estimation of PDF, expected value and variance. 2. Refresher correlation. Calculating with correlation functions. 3. Random processes, correlation function, stationarity, wide sense stationarity, estimation of correlation function (Matlab exercise). 4. Random signal processing, power spectral density function, white noise. 5. AR processes, linear prediction: theory and Matlab exercise. 6. Markov chains. PART II: A course in advanced statistics about linear models, Bayesian inference, classification problems, Gaussian processes and Markov Chain Monte Carlo.	
Study Goals	PART I: 1. Probability Theory - Conditional) probabilities, the law of total probability, and Bayes rule. - Solve probability problems that require the use of axioms of probability. 2. Definition and Description of Random Variables and Processes PDF, PMF, CDF, Covariance, Correlation- Determine if a given PDF, PMF, CDF, variance, (auto/cross-)correlation(-function), (auto/cross-)covariance(-function), power spectral density complies with (theoretical and analytical) requirements. - Convert the description of a probabilistic problem into a probabilistic model using PDF, PMF, or CDF. 3. PDF/PMF and Expected Value Calculate the various forms of expected value of (combinations of) random variables and random processes - For a given (amplitude continuous/discrete and time continuous/discrete) probability model calculate the following probabilistic (marginal, joint and conditional) characterizations: PDF, PMF, CDF, probability of an event, expected value, variance, covariance, correlation, correlation coefficient, auto/crosscorrelation function, auto/crosscovariance function, (cross) power spectral density. - Calculate the PDF, PMF, expected value and variance of a derived random variable. 4. Properties of Random Processes - Independence, orthogonality, uncorrelated, whiteness, IID- Determine if random variables/processes have the following properties: independent, orthogonal, uncorrelated, white, Poisson, Gaussian, Bernoulli, Markov, IID, stationary, WSS, ergodic. - Calculate the expected value, variance, auto/crosscorrelation(function), auto/crosscovariance(function), power spectral density of a linear combination of random variables and of a linearly filtered (WSS, amplitude discrete/continuous, time discrete/continuous) random process. 5. Large NumbersCentral limit theorem, law of large numbers - Solve problems that require the use of the central limit theorem in an engineering context - Explain the law of the large numbers in an engineering context. 6. Statistical Estimators - Estimated mean, variance, and correlation function - Given a set of outcomes, sample functions or realizations, calculate estimators for expected value, variance, and (auto-)correlation function. 12. Application to Engineering Problems and Simulations - Select and translate a simple electrical engineering or computer science problem into mathematical probability model. The emphasis is on problems in signal and image processing, telecommunication, and media and knowledge technology. The class of probability models encompasses the following random variables/processes: Bernoulli, exponential, binomial, Poisson, Gaussian, uniform. - Justify and reflect on the approach taken in calculating or simulating (MatLab) the following probabilistic properties: PDF, PMF, expected value, variance, autocorrelation function, autocovariance function.	

	PART II: After finishing this course, the student is able to apply and derive statistical methods from both the frequentist and Bayesian perspective for - linear models - classification problems - clustering problems - Gaussian process regression The student is able to give a clear presentation about the underlying statistical theory. The student is able to compute several statistical characteristics by hand.
Education Method	PART I: Lectures, working groups (problem solving), laboratory work (a Matlab exercise) Workload is around 15 hours for attending lectures, 5 hours of reading study material and preparing lectures, 15 hours for the lab course, 20 hours for preparing the exam, 3 hours for the exam, and 8 hours for a final report (66 hours in total). PART II: Classes and weekly exercises.
Books	PART I: R.D. Yates and D.J. Goodman, "Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers", ISBN 0-471-17837-3, John Wiley and Sons, New York, 2005, Second Edition. PART II: Simon Rogers and Mark Girolami "A first coruse in machine learning, 2nd edition" Chapman & Hall From this book chapters 1--4, 8 and 9 will be covered.
Assessment	The final grade is the average of the grades you get for part (I) and (II). For part (I) there is a lab and written exam, where the grade is determined by the exam, and the lab assignment should be Passed. If you fail the lab assignment, you'll get a second chance to submit around the time the resit. For part (II), there will be an on-campus written exam. To pass the course, the average should be 5.8 or higher, and the grade of each individual part should be a 5.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.
Exam Hours	PART I: Online exam of 3 hours. PART 2: On campus 3 hour written exam PART II: Written exam of 3 hours.
Permitted Materials during Tests	PART I: Self made notes on a two-sided written A4 sheet. Calculator. PART II: none
Remarks	PART II: This course is particularly interesting for students that are interested in statistical exploratory and quantitative techniques to analyse multivariate data.

CS4176	Algorithms for network-based bioinformatics	5
Responsible Instructor	Prof.dr.ir. M.J.T. Reinders	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	Exam by appointment	
Course Language	English	
Expected prior knowledge	The student is expected to have a basic knowledge of molecular biology, statistics and linear algebra. It is advisable to have followed CS4220 Machine Learning 1 (old code: IN4085 Pattern Recognition).	
Course Contents	Molecular biology is concerned with the study of the presence of and interactions between molecules, at the cellular and sub-cellular level. In bioinformatics and systems biology, algorithms and tools are developed to model these interactions, with various goals: predicting yet unobserved interactions, assigning functions to yet unknown molecules through their relations with known molecules; predicting certain phenotypes such as diseases; or just to build up biological knowledge in a structured way. Such interaction models are often best modelled as networks or graphs, which opens up the possibility of using a large number of readily available algorithms for inferring networks, performing simulations of biology, optimising paths or flows through networks, graph-based data integration and graph mining. Many of these algorithms can be applied (sometimes with slight alterations) to solve a particular biological problem, such as modeling transcriptional regulation or predicting protein interaction/complex formation, but also to derive systems behaviour by breaking down networks into modules or motifs with certain characteristics. In this course, we will first give a brief overview of molecular biology, the advent of high-throughput measurement techniques and large databases containing biological knowledge, and the importance of networks to model all this. We will highlight a number of peculiar features of biological networks. Next, a number of basic network models (linear, Boolean, Bayesian) will be discussed, as well as methods of inferring these from observed measurement data. Building on the network inference methods, a number of ways of integrating various data sources and databases to refine biological networks will be discussed, with specific attention to the use of sequence information to refine transcription regulation networks. Finally, we will give some examples of algorithms exploiting the networks found to learn about biology, specifically for inspecting protein interaction networks and for finding active subnetworks.	
Study Goals	After succesfully completing this course, a student is able to: list the basic elements of a living cell and their interactions, and describe how these can be measured; explain what type of mathematical model is applicable to what measurement(s), at what level(s), in a given systems biology problem; read and comment upon recent network-based computational biology literature; discuss the state-of-the-art in systems biology and integrative bioinformatics, and future challenges	
Education Method	The course consists of a mixture of lectures by the teachers and paper presentations by one or more of the students. Each paper presentation will be followed by a in-depth discussion. There will also be a practical session allowing students to get hands-on experience with network models.	
Literature and Study Materials	Slides, collection of papers and lab course manual (Brightspace).	
Assessment	Students are required to write a proposal for a research project in which they clearly state the biological problem to be solved, the necessary data, the computational approach as well as the innovative parts of the approach. This proposal will be graded. Next, the paper presentations as well as discussions will be graded. The final grade for the course will be based on all these grades: proposal 80% and the presentations/discussions (20%). disclaimer: information may change depending on the developments around the coronavirus.	
Remarks	As students depend on each other (to present the material to the class), a commitment to follow the course through to the end is required.	

CS4220	Machine Learning 1	5
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

CS4230	Machine Learning 2	5
Responsible Instructor	M. Loog	
Instructor	Dr.ir. J.H. Krijthe	
Instructor	Dr. F.A. Oliehoek	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/4/4	
Education Period	3 4	
Start Education	3	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course is the more advanced and research oriented follow-up to CS4220 [Machine Learning 1]. The content of the latter is, therefore, expected as prior knowledge.	
Course Contents	The course will treat a number of machine learning theories and techniques in detail and on an advanced level. Possible topics : - learning theory - Bayesian networks - online learning - Rademacher complexity - Markov decision processes - semi-supervised learning - reinforcement learning - active learning - causal reasoning and discovery	
Study Goals	After successfully completing the course, the student is able to apply the techniques and theories that have been covered in the course. In addition, they are able to develop learning strategies for new and previously unseen situations. Moreover, the student can provide reasoned justifications for these strategies based, for instance, on theory and/or experiment.	
Education Method	Lectures + Q&A sessions	
Assessment	Grading is based on two parts. Following the lectures -- we have about 11 of those, there is an individual assignment that will be graded pass/fail. In addition, there is a written examination that will be graded on a scale from 1 to 10. You pass the course when you both have a pass for the assignment and a passing grade for the written exam. Upon passing the course, your final grade will be the grade for the exam. Finally, note that there is a resit for the written examination, but not for the report. disclaimer: information may change depending on the developments around the coronavirus.	

CS4240	Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	Seminar Computer Vision by Deep Learning	
Expected prior knowledge	Basic pattern Recognition, Basic Machine learning, Basic statistics, Basic probability theory, Programming experience (python + numpy). Note: Without some affinity/experience with python (and numpy) programming it might still be possible to pass the course, but will become quite difficult.	
Course Contents	In this course we will look at a specific field of Artificial Intelligence and Machine Learning: Deep learning. Deep learning has shown remarkable success with large data sets and unstructured input data such as raw images/audio/text. Topics include: feed forward networks, back-propagation, optimization, convolutional nets, recurrent nets, self-attention, unsupervised methods. The course will have lectures, a seminar, a lab practical and a project: - The lectures will be on generic topics, following the book; building the backbone. - The lab assignments will have you apply basic concepts of the lecture in python notebooks. - The seminar will have students read, critique, and present relevant deep learning research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The project will have students apply and design their own (small) deep learning project in the context of scientific reproduction.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the different deep learning techniques reviewed in the course, such as SGD, MLPs, CNNs, RNNs, GANs. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [LO4]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (eg: Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) problem domain which may need to re-implement, run, evaluate, investigate, extend existing research or code [LO8] Write clearly and concisely about your code, method, results, and analysis.	
Education Method	Lectures for basic theory based on the literature Assignments: we help you to become familiar with PyTorch; applying concepts from the lecture on small problems. Lab project: design and execute your own deep learning project in the context of scientific reproductions using https://reproducedpapers.org/ Seminar: paper reading, critiquing, and presenting.	
Literature and Study Materials	Books: freely available online: - http://www.deeplearningbook.org/ - https://d2l.ai/ Research papers will be made available through Brightspace. Assignments are based on PyTorch: https://pytorch.org/	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Project: in a small group of students you work on a deep learning paper reproducibility project. 3. Exam about the papers, assignments, and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4245	Seminar Computer Vision by Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	S. Pintea	
Contact Hours / Week x/x/x/x	0/0/0/8	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Required for	MSc thesis in the Computer Vision lab	
Expected prior knowledge	Deep Learning (CS4240)	
Course Contents	The recent boom in computer vision and automatic image understanding represents an inflection point in human productivity, permeating wide aspects of the economy and society. Examples of visual tasks which are repetitive or require expert knowledge include medical diagnosis, industrial inspection, autonomous vehicles, etc. When machines can meaningfully assist or even completely take-over such tasks it will change the world as we know it. The breakthrough in the 2012 ImageNet automatic image recognition competition shows all previously existing methods decisively defeated by a deep neural network. Deep learning replaces feature engineering methods and is able to successfully learn image features from huge annotated datasets. This course is on automatically understanding visual content such as images and videos by deep learning. Topics include: Fundamentals in Vision, object detection, per-pixel labelings, video recognition, image similarity learning, efficiency, self-supervision, 3D computer vision, adverserial attacks, explainability, generative models. The course will have lectures, a seminar and a lab practical: - The lectures will be on established topics based on the current literature. - The seminar will have students read, critique, and present relevant computer vision research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The lab will have students apply and design their own (small) computer vision project. The course build on top of the Deep Learning course (CS4240) and follows a similar setup.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the deep learning techniques reviewed in the course for computer vision applications such as image classification, object detection, per-pixel labelings, video recognition, image similarity learning. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [L04]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (we focus on Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) Computer Vision problem. [LO8]. Write clearly and concisely about your code, method, results, and analyis.	
Education Method	Lectures Lab project: design and execute your own Computer Vision project. Seminar: paper reading, critiquing, and presenting.	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Students will have to submit relevant questions about papers/lectures 3. Lab assignment: in a small group of students you work on a deep learning project. 4. Exam about the papers and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4250	Selected Topics in Molecular Biology	5
Responsible Instructor	Dr. T.E.P.M.F. Abeel	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1	
Course Language	English	
Course Contents	This course covers a broad range of essential topics in molecular biology necessary to start critically working with bioinformatics algorithms. The choice and implementation of algorithms critically depend on a proper understanding of the application domain. In this case the application domain is biology and in this course we review key biological concepts and topics that have impact on the implementation and selection of various computer algorithms and models. We use the following book: B. Alberts et al., 'Molecular Biology of the Cell', 6th edition Publication Date: November 18, 2014 ISBN-13: 978-0815344322 ISBN-10: 0815344325 A detailed overview of the actual reading material will be provided through the online learning platform Brightspace and will include material from Chapters 1,3,4,5,6,7 and 8. Extra material to aid in learning are available from the publisher's website. Covid-19 disclaimer: all information here is correct at the time of writing. This may change based on developments around COVID-19. Students will be informed as soon as possible through Brightspace.	
Study Goals	The goal of this course is to learn about the basic concepts in molecular biology required for bioinformaticians. At the end of this course: students will be able to reproduce, discuss and reflect on information about basic molecular biological processes, concepts and ideas. This includes: - cell structure, organization, information encoding and transfer - core concepts around DNA, chromosomes, genomes - DNA replication, repair and recombination - central dogma molecular, including replication, transcription, translation - regulation of the processes above	
Education Method	Self-study with weekly Q&A sessions.	
Books	B. Alberts et al., 'Molecular Biology of the Cell', 6th edition Publication Date: November 18, 2014 ISBN-13: 978-0815344322 ISBN-10: 0815344325	
Assessment	Oral exam and assignments by appointment disclaimer: information may change depending on the developments around the coronavirus.	

CS4255	Algorithms for sequence-based Bioinformatics	5
Responsible Instructor	Dr. T.E.P.M.F. Abeel	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	2	
Start Education	2	
Exam Period	2	
Course Language	English	
Course Contents	Covid-19 disclaimer: all information here is correct at the time of writing. This may change based on developments around COVID-19. Students will be informed as soon as possible through Brightspace. Bioinformatics analyses in genomics aim to compare large sets of genomes in order to understand and explain differences in traits of an organism. Contemporary methods are powered by fundamental algorithms and data structures, which are efficient and scale to large data sets. A thorough understanding of these algorithms and data structures is necessary for advanced users and developers in this area. In addition, understanding how comparative genomics is developing is important to shape your own research. In this course, we will cover genome analysis, variant analysis, and pangenomics. Core concepts, applications, and future trends will be discussed, with a focus on the algorithms and data structures underlying state-of-the-art methods.	
Study Goals	After having followed this course, the student has a good understanding of algorithms and data structures in genomics used for DNA sequence analysis. The student is able to implement algorithms in python, and can translate methods described in scientific literature into a working implementation.	
Education Method	The course is offered as a mix of lectures, exercises and a project	
Books	http://bioinformaticsalgorithms.com	
Assessment	60% graded exercises, 40% project. A resit opportunity is available for both components Students must receive a 'pass' mark for each component individually (>=5.8). Notes: - No scores and/or submissions are transferred to the next year or from previous years. - Rules and assessment format may change due to COVID19-regulations. This will be communicated as soon as possible. disclaimer: information may change depending on the developments around the coronavirus.	

CS4260	Machine Learning in Bioinformatics	5
Responsible Instructor	Dr. J.S. de Pinho Gonçalves	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures + 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	

Expected prior knowledge

Recommended for fundamental knowledge in machine learning and multivariate data analysis:
CS4220 Machine Learning 1
CS4070 Multivariate Data Analysis

Recommended for basic concepts of molecular biology:
CS4250 Selected Topics in Molecular Biology

Required:
CSE1200/TI1106M Calculus (or similar)
CSE1205/TI1206M Linear Algebra (or similar)
CSE1210/TI2216M Probability Theory and Statistics (or similar)
The project requires basic programming skills.

Course Contents

Learning from patterns in molecular biology data plays an important role in diagnosing disease, discovering new targets for therapy, and more generally in answering biological questions that lead to an improved understanding of biological systems with relevance to human health and industrial areas including biotechnology and agriculture.

This course focuses on methodology for the analysis of high-dimensional data in molecular biology, naturally addressing challenges that commonly arise in the field such as learning from unlabelled data or from small numbers of samples. The methodology is introduced in the context of meaningful applications in molecular biology with examples using real data.

Covered topics will be (a selection of) the following:
- Discrete probability models and statistical modeling: statistical models for experiments with categorical outcomes, goodness of fit, maximum likelihood and Bayesian estimation.

- Mixture models: finite mixture models of normals and the EM algorithm, common infinite mixture models such as beta-binomial and gamma-Poisson, ECDF and bootstrapping.
- Clustering: comparing observations, iterative partitioning, density-based clustering, hierarchical clustering, clustering validation, choosing the number of clusters.

- Statistical hypothesis testing: p-values, single and multiple hypothesis testing, family-wise error rate, (local) false discovery rate.
- Testing using high-throughput count data: multifactorial designs, linear models, analysis of variance, generalized linear models, robustness and outlier detection, shrinkage estimation.

- Linear dimensionality reduction (PCA): preprocessing data for multivariate analysis, projecting onto lower dimensions, matrix decomposition (SVD), biplot representations, projecting additional variables for interpretation).
- Multivariate methods for heterogenous data: orderings, gradients and latent variables (ordination); multidimensional scaling (MDS), robust (non-metric) MDS, batch effect removal, correspondence analysis, finding gradients and trajectories (local non-linear methods such as tSNE and UMAP), canonical correlation.

- Supervised learning: discrimination, performance measures, curse of dimensionality, generalizability and model complexity, regularization and penalization, cross-validation, supervised learning methods (SVM, decision trees, ...), method hacking.

- Design of high-throughput experiments and their analyses: types of variability (error, noise, bias), confounding, dependencies, batch effects, statistical power, mean-variance relationships and data transformations, workflow design, data representation, efficient computation.

Study Goals

After successfully completing this course, the student should be able to:
- Recognise, characterise, and interpret different kinds of high-throughput molecular biology data and their statistical properties.
- Recognise, categorise and compare common statistical techniques and machine learning algorithms and the data analysis problems that they address.
- Recognise typical research questions that can arise when analysing such kinds of data, reason about and select appropriate methodology to address them.
- Understand, reason about, and discuss the different steps of a data analysis workflow: from experiment design to the interpretation of results.
- Design a research plan, execute it, and write a scientific paper about it.

Education Method

The course is run in a flipped classroom setting.

Lectures: students take turns explaining the material from the different chapters of the course book. All students are required to read and prepare the course material before every lecture, so that they can contribute to the discussion of the material in the classroom.

Project/labs: students will work in small groups on a research project throughout the course. Each group will present and discuss the status of the project weekly or bi-weekly, and deliver a written report at the end.

There will be one or two lectures (90 min. each) and one project discussion lab (45 to 90 min.) per week. The frequency and duration of the sessions will depend on the number of students following the course.
The exact schedule will be determined after the first lecture.

Books

The material of the course follows the book:
"Modern Statistics for Modern Biology"
Authors:
Susan Holmes, Stanford University, California
Wolfgang Huber, European Molecular Biology Laboratory
Published: February 2019
ISBN: 9781108705295

The complete book can be browsed online at: http://web.stanford.edu/class/bios221/book/.
This website also contains the R code and data accompanying the examples in the book.

Assessment	- 25% Book chapter presentation - 15% Participation - 60% Project report/paper Students are graded individually. A passing grade for this course can only be obtained if the grade of every component except participation is at least 5.8. A resit opportunity is available for the project component. Grades of individual components are not carried over to future editions of the course. Disclaimer: information may change depending on developments around COVID-19.

CS4329	**Recent topics in bioinformatics**	**5**
Responsible Instructor	Prof.dr.ir. M.J.T. Reinders	
Instructor	Dr. J.S. de Pinho Gonçalves	
Instructor	Dr. T.E.P.M.F. Abeel	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Bioinformatics is at the heart of many modern systems biology analyses, and encompasses the application of statistics and computer science to (large-scale) biomolecular datasets. In essence, bioinformatics is about smart ways of extracting knowledge from the enormous amounts of data that can be generated using modern measurement techniques. For instance, it plays an important role in finding the genetic origins of various diseases, such as cancer, diabetes or alzheimer. Covid-19 disclaimer: all information here is correct at the time of writing. This may change based on developments around COVID-19. Students will be informed as soon as possible through Brightspace.	
Study Goals	After successfully completing this course, the student is able to: explain several high-throughput data acquisition experiments, such as DNA/RNA sequencing, and discuss the benefits and limitations of these methods comprehend the statistical and computer science issues in analyzing high-throughput data discuss the basic systems biology approach, and the role of high-throughput measurements, gene selection and classification therein explain bioinformatics methods, algorithms and models to a non-expert audience implement or execute basic algorithms from descriptions provided in scientific literature read and comprehend a current scientific paper and reflect on the bioinformatics methods used in such paper	
Education Method	In this course we will study some key examples of bioinformatics analyses by reading a set of selected papers that present some significant biological conclusions.The course is run a flipped class-room students take turns guiding us through the selected papers. Teachers moderate the discussion and fill in any gaps in understanding.All students are required to read, and prepare the course material before every lecture to effectively participate. Each week there are one or two lectures, each of 90 minutes In each lecture one paper (the course material) will be discussed in detail. One or more students will present and explain the details of this paper. It is essential that you highlight the (bioinformatics) methodology of the paper. The schedule for this will be prepared in the first lecture. This presentation is graded. All students are expected to have read the paper and should have an active role in the discussion about the paper. Each student should be prepared to raise at least two critical remarks or questions during the discussion. The quantity and quality of participation of each student is graded. The student will perform a practical project in groups where they gain practical experience in state-of-the-art bioinformatics methods.	
Assessment	- Presentation (20%) - Participation during discussions (20%) - Reporting about unseen paper with oral presentation (60%) disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Programming Languages 2021

CS4130	Seminar Programming Languages	5
Responsible Instructor	E. Visser	
Responsible Instructor	Dr. S.S. Chakraborty	
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Dr. J.G.H. Cockx	
Instructor	C.B. Poulsen	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Followed at least one other Programming Languages master course	
Course Contents	Programming languages is a core field in computer science that studies the design, theory and applications of both new and existing programming languages. Topics in programming languages include compiler construction, program analysis, program transformations, meta programming, parsing, formal semantics, program verification, and type systems. In this course, we will read scientific journal and conference articles in the field of programming languages to get a deeper understanding of programming languages. If you wish to do a MSc thesis in the programming languages group, we highly recommend taking this course.	
Study Goals	The student will acquire: - Skills to read and discuss scientific articles. - Understanding of the topics in the research field of programming languages. - Understanding of the research methodology in the research field programming languages.	
Education Method	We will run this seminar as a discussion seminar with meetings twice a week. In each meeting, we discuss a scientific article that has been studied by the participants in advance. The following activities are required for each meeting: - Reading a scientific article - Writing and submitting a short summary of the article (max 0.5 pages) - Active participation in the discussion of the article Expected Workload: - 4h Discussion sessions - 6h Reading paper at home - 2h Writing summary at home	
Literature and Study Materials	Papers from the programming languages literature will be assigned at the start of the course	
Books	No books	
Assessment	Students get a grade for each meeting based on the participation in the discussion. The final grade is the average of the grades for the meetings. There will not be a resit for the course. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	not applicable	
Enrolment / Application	The number of participants for this course is limited. Students in the second year of the master and students that follow the Programming Languages specialization have priority.	
Judgement	The final grade is the average of the grades for the meetings.	

CS4135	Software Verification	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Ir. K.F. Wullaert	
Contact Hours / Week x/x/x/x	0/0/2/0 + 0/0/4/0 practicum	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	How can we ensure that software cannot crash and is guaranteed to be correct? In this course we tackle this question by viewing programs and programming languages as mathematical objects. That way we can use logic to prove properties about programs and thereby guarantee that software is correct. To make reasoning about actual programs and programming languages feasible, we will not be doing these proofs by hand, but instead use a tool called a proof assistant to build proofs that can be checked by a computer. As we will show during this course, proof assistants turn the activity of doing proofs into programming. This course assumes familiarity with functional programming and elementary logic. This course is a specialization course for programming languages and software engineering	
Study Goals	After this course students will be able to: - State and prove properties of functional programs in logic. - Specify the semantics of a programming language in logic. - State and prove the correctness of imperative programs. - Use a proof assistant to perform a mechanized proof.	
Education Method	This course consists of a weekly lecture of 2 hours and a lab session of 4 hours. During the lab sessions students will work on proving simple theorems. Towards the end of the course students will carry out research projects that apply the ideas of the course.	
Literature and Study Materials	Supplementary material: Free online text book "Logic and Proof": https://leanprover.github.io/logic_and_proof/ Free online text book "The Hitchhikers Guide to Logical Verification": https://github.com/blanchette/logical_verification_2021/raw/main/hitchhikers_guide.pdf	
Assessment	The final grade consists of the following parts: - A programming project in a proof assistant. - A written exam Both have weight 50% and both should be 5 or higher. The weighted average should be 5.8 or higher. The research project should be done individually.	
Co-Instructor	E. Visser	

CS4200-A	Compiler Construction	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	6/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	- programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-A covers the following topics: * Syntax and parsing - concrete syntax, abstract syntax - context-free grammars - derivations, ambiguity, disambiguation, associativity, priority - parsing, parse trees, abstract syntax trees, terms - pretty-printing - parser generation - syntactic editor services * Static semantics and type checking - name binding, name resolution, scope graphs - types, type checking, type inference, subtyping - unification, constraints - semantic editor services	
Study Goals	After this course, students should be able to: * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define the syntax of a programming language using declarative syntax definition that describes the concrete and abstract syntax of a programming language * Define basic editor services * Define the type system (name binding and typing rules) of a programming language using constraint generation rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the front-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%) - course project (50%)	

	Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	No materials are permitted during the exam.
Judgement	to be decided

CS4200-B	Compiler Construction B	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/6/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	- CS4200-A: Compiler Construction A (recommended) - programming (required) - software engineering (recommended) - concepts of programming languages (recommended) - formal languages and automata (recommended)	
Course Contents	Note that the title of this course should be "Compiler Construction B", not "Compiler Construction Project". The course combines theory and practice of compiler back-ends. Compilers translate the source code of programs in a high-level programming language into executable (virtual) machine code. Nowadays, compilers are typically integrated into development environments providing features like syntax highlighting, content assistance, live error reporting, and continuous target code generation. This course studies the architecture of compilers and interactive programming environments and the concepts and techniques underlying the components of that architecture. For each of the components of a compiler we study the formal theory underlying the language aspect that it covers, declarative specification languages to define compiler components, and the techniques for their implementation. The concepts and techniques are illustrated by application to small languages or language fragments. The course consists of two courses CS4200-A in Q1 and CS4200-B in Q2. In CS4200-A, we study the 'front-end' of the compiler. In CS4200-B, we study the 'back-end' of the compiler. In the homework assignments of the course students practice their understanding of the theory by solving small problems. In the lab assignments of the course students build a complete compiler and programming environment for a small language, divided over front-end in CS4200-A (Q1) and back-end in CS4200-B (Q2). Course CS4200-B covers the following topics: * Transformation - rewrite rules, rewrite strategies - simplification, desugaring * Dynamic semantics and code generation - operational semantics, program execution - virtual machines, assembly code, byte code - code generation - memory management, garbage collection * Data-flow analysis - control-flow, data-flow - monotone frameworks, worklist algorithm	
Study Goals	After this course students students should be able to * Describe the architecture of a compiler and programming environment and the role of each component of that architecture * Explain the basic concepts of the formalisms for the definition of the components of a compiler and apply these formalisms to define small languages * Explain the algorithms and techniques for the implementation of compiler components and apply these techniques to examples * Define transformations on abstract syntax terms to simplify programs * Define a code generator that translates source language abstract syntax trees to object language instructions using rewrite rules * Define data-flow analyses using control-flow and data-flow rules * Construct tests for each of the components of a compiler in order to determine its correctness * Integrate the components into a working compiler and programming environment	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab session (which may start with a group tutorial) - reading lecture material and papers - making homework assignments - building the back-end of a compiler	
Literature and Study Materials	Lecture slides and selected papers from the literature We will use the Spoofax language workbench (http://metaborg.org) for the course project and for some of the homework assignments. Reading material and homework assignments will be published on the course website; see http://pl.ewi.tudelft.nl/education/compiler-construction/ We will use WebLab (https://weblab.tudelft.nl/cs4200/) for the submission of homework assignments and GitLab for the submission of project work.	
Assessment	The final grade will be determined by the following components - final exam (50%)	

	- course project (50%) Separate grades are given to each of these components, but only the final grade will be registered in Osiris. The grades for each of the components should be at least 5.0 and the final grade should at least be 5.8. disclaimer: information may change depending on the developments around the coronavirus.
Permitted Materials during Tests	not applicable
Judgement	to be decided

CS4275	Web Programming Languages	5
Responsible Instructor	E. Visser	
Instructor	Dr.ir. D.M. Groenewegen	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	Languages and frameworks for web programming are constantly evolving. Over the past decade, there has been a large shift from applications with server-side rendering of separate web pages, to single page applications with client-side rendering and web service back-ends. One of the strengths of web programming technologies is separation of concerns. The concerns such as describing content, styling, behavior, and persistence, are often separated with their own domain-specific languages. A particular programming problem that newer web programming languages tackle is dynamic user interfaces with automatic fine-grained updates. This problem is not restricted to web applications, but applies to any GUI programming abstraction. Consequently, the technologies for web programming are also relevant for development of cross-platform mobile and desktop applications. In this course, students will analyze web languages and frameworks from a programming language perspective. They will explore the underlying concepts and abstractions, and learn how the tools relate to each other. The investigated web technologies range from more traditional full-stack web development solutions with persistence and templating, to popular client-side UI solutions with fine-grained updates and state synchronization.	
Study Goals	The course gives students the conceptual and technical skills to understand the role of programming languages in web programming and the advantages and disadvantages of different approaches.	
Education Method	Attending the course involves - attending weekly lectures - attending weekly lab sessions - reading lecture material and papers - making project assignments	
Assessment	Students get a grade for each of the project assignments. The final grade is the weighted average of the grades for the project assignments. There will not be a resit for the course. Disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is the average of the grades for the project assignments.	

CS4280	Language-Based Software Security	5
Responsible Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course has no formal prerequisites. However, for the homework assignments you will have to implement several program analysis techniques using the Scala programming language. If you have not used Scala before, you are thus expected to learn the basics of the language through self-study.	
Course Contents	Security vulnerabilities often arise due to programming errors in the source code of an application. Recent programming errors with severe security implications include Heartbleed (buffer over-read), Shellshock (code injection), and goto-fail (ill-formated code). Rather than hunt for individual vulnerabilities in programs, a more structural approach to improve security is to improve the programming language. This is the goal of language-based security: to rule out whole classes of potential security vulnerabilities in one go. This course studies various security properties and program analysis techniques for enforcing these properties at the level of the programming language to improve software security. In particular, we will study the following properties: - Memory safety: prevent buffer overflows and overreads - Type safety: prevent undefined behaviour - Information flow control: prevent data leaks and code injection attacks We will study techniques to address these problems at the language level through dynamic analysis, static analysis, and language design. To facilitate a precise study and comparison, we will define the above techniques formally in class. To facilitate student experimentation and exploration of trade-offs, students will implement the above techniques in homework assignments.	
Study Goals	After taking this course, students should be able to: 1. Describe the nature and causes of security vulnerabilities in software systems, and give concrete examples of how these security vulnerabilities can be exploited. 2. Explain the properties that can be enforced at the level of the programming language to rule out security vulnerabilities, such as memory safety, type safety, and non-interference. 3. Formally define the semantics of a simple programming language. 4. Formally define dynamic and static analysis techniques for enforcing these security properties. 5. Implement these techniques for a small programming language. 6. Discuss and evaluate the importance of soundness and precision of a given program analysis. 7. Contrast programming languages based on the set of countermeasures they provide, and give an appropriate recommendation for a specific application. 8. Analyse and apply results from scientific literature in the area of language based security.	
Education Method	The course work consists of the following activities: 1 or 2 instruction sessions per week. Weekly homework assignments consisting of theoretical questions, programming assignments, and reading assignments	
Assessment	The assessment for this course consists of two parts: The weekly homework assignments will test your ability to design an implement (variants of) the techniques discussed in the lectures (study goals 3-5). This counts for 40% of the total grade. The final written or oral exam will test your theoretical understanding of the security vulnerabilities and their countermeasures discussed in class (study goals 1-2) and your ability to discuss and contrast the different aspects of these techniques (study goals 6-8). This counts for 60% of the total grade. To pass the course, each of these grades (homework assignments and final exam) should be 5.0 or higher, and the final grade should be 5.8 or higher (and will be rounded to the nearest half grade point). disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	E. Visser	

CS4405	Analysis of Concurrent and Distributed Programs	5
Responsible Instructor	Dr. B. Özkan	
Responsible Instructor	Dr. S.S. Chakraborty	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software systems are becoming highly concurrent and distributed to utilize modern multicore architectures and increasing speed and bandwidth in networks. Shared-memory concurrency in multicore programs and message-passing concurrency in distributed programs share many common abstractions and problems. In the multicore era, all performance-critical software employs some form of concurrent programming; typically shared memory concurrency. In this setting, programmers use a number of primitives to develop efficient and correct concurrent programs. To do so the programmers have to understand the behaviors of the primitives and reason about them. It is also important to match the programming paradigms and underlying architectures. For instance, traditionally programmers have assumed that a multithreaded program executed simply by interleaving the executions of its threadsa model known as sequential consistency (SC). This assumption is, however, invalidated both by mainstream multicore architectures, which often execute instructions out of order, and by compilers, whose optimizations affect the outcomes of concurrent programs. As a result, concurrent programs have more outcomes than SC allows. In the distributed setting, the units of concurrency are independent processes that do not share memory but communicate by exchanging asynchronous messages. The execution of such a system involves two main sources of nondeterminism: concurrency and partial failures. As the processes run concurrently, the exchanged messages can be delivered and processed in many different orderings. The distributed set of processes is also prone to network of process failures. The trade-off between the systems availability in the existence of failures and the consistency between the processes gives rise to a spectrum of weak consistency notions. It is important to reason about concurrency, possible failures, and consistency guarantees to implement distributed programs correctly and understand their behavior. This course aims to explore analysis techniques for concurrent and distributed programs. Outline of Lectures: Shared memory concurrency: - Abstractions for shared memory concurrency - Relaxed memory concurrency - Correctness of concurrent programs Distributed concurrency: - Distributed system components, models and assumptions - Fundamental abstractions for distributed systems	
Study Goals	This course aims to give students a deep understanding of concurrency and distribution in modern systems and hands-on experience for analyzing these systems. At the end of the course, the students will be able to: - Analyze and reason about concurrent and distributed programs - Apply and analyze existing techniques on unseen problems - Be able to pursue independent further research in the area	
Education Method	The course consists of the following education methods: - Lectures for reviewing concurrency and distribution concepts - Homeworks/assignments - Developing a course project, writing a report, and presenting it (course project) To finish the course, students (in teams) will have to: - Study several papers which will be discussed during the lectures - Deliver their assignments - Deliver and present their implementation project	
Assessment	The final grade is composed of: research project implementation) (40%) + research project report (20%) + research project presentation (20%) + homework assignment (10% + 10%). No written exam. Resits are not offered.	

CS4410	Category Theory for Programmers	5
Responsible Instructor	Dr. B.P. Ahrens	
Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Categorical structures occur in programming languages on different levels: (1) within programming languages, providing design principles and guidance on how to write modular and correct-by-design programmes (as demonstrated in the practical programming language Haskell) and (2) in the design and study of programming languages, as a guiding meta-theory. In particular, category theory provides a mathematical justification for recursion schemes for inductive datatypes. This course aims to provide solid foundations on both (1) and (2).	
Study Goals	- Use categorical constructions (e.g., monads) in the design and structuring of computer programmes in Haskell - Prove properties of computer programmes, guided by categorical intuition - Understand categorical fusion laws and how to use them to optimize code - Understand the theory of infinite data structures and apply it to practical problems	
Education Method	Learning in this course is achieved through lectures, problem sessions, and guided self-study.	
Assessment	Exam at the end of the term, counts for 100% of the mark.	

IN4333	Language Engineering Project	5
Responsible Instructor	E. Visser	
Contact Hours / Week x/x/x/x	0/0/0/4 (lab)	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Compiler construction CS4200-A and CS4200-B.	
Course Contents	"Software systems are the engines of modern information society. Our ability to cope with the increasing complexity of software systems is limited by the programming languages we use to build them. Bridging the gap between domain concepts and the implementation of these concepts in a programming language is one of the core challenges of software engineering. Modern programming languages have considerably reduced this gap, but often still require low-level programmatic encodings of domain concepts. Or as Alan Perlis formulated it in one of his famous epigrams: 'A programming language is low level when its programs require attention to the irrelevant'. A fixed set of (Turing Complete) programming constructs is sufficient to express all possible computations, but at the expense of considerable encoding that obfuscates the concepts under consideration. Linguistic abstraction can be used as a tool to capture our emerging understanding of domains of computation." (Visser, SCP 2015) In the precursor compiler construction course (CS4200), students learn the basics of language engineering by building a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts. In the precursor compiler construction course (IN4303), students learn the basics of language engineering and build a complete definition for a small programming language. In this course, students learn to apply language engineering principles and tools to a real (domain-specific) programming language in a new project, i.e. without following a path set out in detail by the instructor. Thus, they will experience that the design of a real programming languages requires trade-offs and compromises. Typically, students implement a previously existing language. But designing a new language is also an option provided there is a good plan that is discussed _before_ the course starts.	
Study Goals	In this course students learn to apply language engineering principles and tools to a real (domain-specific) programming language. Explore the definition of all aspects of a programming language: syntax, name binding, type analysis, transformations, code generation.	
Education Method	This is a project course. Students deepen their language engineering skills and insights by building a complete language definition. Students work in teams of two on the definition of a (domain-specific) programming language using the Spoofax Language Workbench. Assistance and feedback is provided during weekly lab hours. The project should span the full life cycle of language implementation including a test suite, IDE, code generator, and distribution of the result as an Eclipse plugin.	
Literature and Study Materials	- Documentation of the design and implementation of a specific language - Papers about language definition techniques	
Assessment	The work is assessed based on a code review of the language definition, a written report about the project, and a presentation in the final project workshop. The course has no resit. disclaimer: information may change depending on the developments around the coronavirus.	
Judgement	The final grade is based on the following components: - git repository with language project (40%) - written report about the project (30%) - presentation (slides) (30%)	

IN4387	System Validation	5
Responsible Instructor	C.B. Poulsen	
Contact Hours / Week x/x/x/x	4,0,0,0 Lectures & 2,0,0,0 lab	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	Embedded Systems Masters	
Expected prior knowledge	There are no strict entry conditions for this course. However, prior knowledge of requirements analysis is recommended. Furthermore, a good basic knowledge about logic and set theory is extremely beneficial.	
Parts	Behavioural specification of sequential and parallel using labelled transition systems, process algebra, and abstract data types; model checking of such systems using the modal mu-calculus. Model-based testing.	
Summary	Everyone who ever designed an embedded system or a communication protocol involving several components executing simultaneously has experienced that such software is inherently susceptible to bugs. Typical problems that occur are race conditions, deadlocks, and unexpected interplay between different components. Due to the parallel nature of these systems, it is notoriously hard to detect such bugs using testing (for example, timing plays a crucial role). The following quote from the famous Dutch computer scientist Edsger W. Dijkstra illustrates a further problem with testing. Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence. Edsger W. Dijkstra In this course, we study model checking, which in contrast to testing can also be used to show the absence of bugs. Model checking is a technique in which we consider all states in (a model of) the system based on an abstract model. Based on this state space we verify whether the model satisfies the desired properties. Properties are typically derived from the requirements of the system. We will restrict ourselves to verification techniques that do not reason about timing (merely about the order in which event happen). Finally, we see how model-based testing can be used to show that an implementation conforms to the specification of the system.	
Course Contents	Behavioral Specification using Process Theory (Labelled Transition Systems, various notions of behavioral equivalence) and process algebra. Model checking the modal mu-calculus, and model-based testing using IOCO.	
Study Goals	Upon completion of the course: 1. The student knows the fundamental theory necessary for specifying the behavior of embedded systems and for reasoning about this behavior. 2. The student can describe simple systems using this theory. 3. The student can formally specify requirements and prove (or disprove) them on the behavior. 4. The student is able to model a concrete embedded system, and verify that it satisfies its requirements. 5. The student is able to show that an implementation of a system conforms to its specification.	
Education Method	Lectures + Programming Assignments + Practical Project The course is structured into two parts: 1. There will be weekly mandatory programming assignments in the first four weeks of course will be a small set of mandatory. The programming assignments are assessed as pass/fail. The programming assignments are due after the first four weeks of the course. 2. In the last four weeks of the course, you will self-organize into groups of (about) 4 students, and will develop and verify a model of an embedded system. You will write a report that documents your model and its development. There will be a written exam with programming assignments at the end of the course.	
Computer Use	The theory introduced in this lecture is at the heart of the mCRL2 tool set. This tool set can be used to specify and verify systems, and visualize them. To be able to carry out the project it is required that the mCRL2 tool set is installed on your laptop (or one of the TU Delft systems, if you do not have a laptop you can use). It is open source software, and is free of charge. The software can be obtained from https://www.mcrl2.org.	
Literature and Study Materials	The course is based on the book by Groote and Mousavi (see "Books"). All other materials will be published on Brightspace.	
Books	J.F. Groote and M.R. Mousavi. Modeling and Analysis of Communicating Systems. MIT Press, 2014. ISBN: 9780262027717 (Chapters 1-7,11 are mandatory)	
Assessment	The result of this course will be based upon the results of the written examination (50%) and the practical project (50%). For both the programming exam and the practical project, a minimum of 5.0 is required in order to pass the course. To be eligible for taking the exam you must submit and pass the mandatory programming assignments for the first four weeks of the course. Grades of the project or written exam do not automatically carry over from previous years, so upon retaking the course talk to your lecturer first. For the exam a resit is scheduled. Please note that the study guide information for this course may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	The exam will be a 3 hour written exam with programming questions. You are allowed to use the book and any other static resources. You are not allowed to communicate or discuss exam questions with anyone but members of the teaching team for the course. Discussing or copying code will be considered fraud, and is reason for expulsion from the course.	
Enrolment / Application	Brightspace	
Co-Instructor	E. Visser	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

QCE/ Network Architectures and Services 2021

EE4396	Mobile Networks	5
Responsible Instructor	Dr. R. Litjens	
Contact Hours / Week x/x/x/x	0/0/0/5	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Not formally required, but recommended to have the basic knowledge taught in 'Fundamentals of Wireless Communications' (ET4358). If you didn't take this course, key aspects will be repeated during the class. In case you are lacking a comfortable background, talk to me in the first week of class and I'll help you catch up.	
Course Contents	The course addresses the technology and management of contemporary and future mobile cellular communication networks. With a focus on radio access, as it dominates both the performance and cost aspects of mobile networking, the course addresses 5G, 4G (LTE/LTE-A), 3G (UMTS/HSPA) and 2G (GSM/GPRS/EDGE) technologies, all of which are operational network technologies. An outlook to 6G will also be given. Fundamental aspects of cellular networking are explained and discussed, including e.g. the intrinsic tradeoffs between coverage, capacity and quality and key aspects that influence performance (traffic, propagation, technology, deployment, spectrum. Cellular network planning, dimensioning and performance analysis are treated, as well as radio resource management mechanisms (adaptive beamforming, channel-aware single/multi-user scheduling, handover control, link adaptation, admission, congestion control,) and the current trend towards (potentially AI/ML-based) self-management. Cutting-edge radio features including single/multi-user massive MIMO beamforming, which is at the heart of 5G, are also covered. Considering the on-going technological developments in this exciting domain, the course material is updated every year. The principal lecturer has a 25+-year track record of research and consultancy in the field. The course is enriched with notes about actual deployments, historical anecdotes, popular misconceptions, fascinating factoids, illustrative demonstrations and (potentially) guest appearances.	
Study Goals	The objective of the course is to provide the student with a thorough understanding of contemporary mobile cellular networking technologies, their fundamental properties and tradeoffs, as well as the key challenges and approaches related to network dimensioning, planning and (self-)optimization towards efficient provisioning of coverage, capacity and quality.	
Education Method	Lectures, demonstrations, in-class discussions, assignment(s).	
Literature and Study Materials	Primarily: the lecture slides! For the enthusiastic student, supplementary background materials are recommended in the form of e.g. (white/scientific) papers and book chapters.	
Books	A number of books on wireless/mobile communications in general or about 2/3/4/5G technologies specifically are recommended as background material; specifics will be given in class / in the lecture slides. As said, this is merely recommended for the most eager students. In principle, the lecture slides cover all relevant content.	
Assessment	If COVID regulations comfortably allow: written, closed book exam; otherwise: oral, closed-book exam. And one (may be two) assignments.	
Co-Instructor	Prof.dr.ir. P.F.A. Van Mieghem	

ET4034	Telecom Business Architectures and Models	4
Responsible Instructor	Dr. E.F.M. van Boven	
Responsible for assignments	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	none	
Summary	Edgar van Boven, (KPN, TUD EEMCS)-- Past, present & future of communications -- Assignment 1 Eric Smeitink (KPN, TUD EEMCS)--Strategy, Spectrum & 5G Ton van der Knaap, (Stedin)-- Process Architecture: A Challenge Samuel Pronk, (Krauthammer)-- Real Options-- Assignment 2 Ramin Hekmat, (KPN)-- Security and privacy in contemporary communication networks-- Assignment 3 John Hoffmans, (KPN)--The principles of telecommunications transport networks Wouter de Vries (KPN)-- Artificial Intelligent Processes for People Tim Daeleman, (Prodapt Consulting)and Erik Lemmens (KPN)-- Digital Telecom Platforms Niels van Adrichem, (TNO)--Future Internet architectures Frank Mertz (KPN)--KPN Field Labs Remco Helwerda (KPN) -- Digitalization Marco van der Pal, (Cap Gemini)--Infrastructure challenges for large enterprises Ruud van de Bovenkamp (KPN)-- Content Delivery Networks -- Assignment 4 Hong Gie Ong (ING) and Jessica Maes (I-Control) -- Opportunities & Threats of the Digital Revolution Pieter Veenstra, (NetNumber)-- Service Architectures, the next generation Edgar van Boven (KPN, TUD EEMCS)-- Value Case pitch practicing -- Assignment 5	
Course Contents	The essence of the course The ET4034 course Telecom, Architectures and Business Models was designed in 2001, initiated by Prof. Dr. Ir. Nico Baken. The aim of the course is giving students from any faculty of the Delft University of Technology the opportunity to learn about the constituents of the telecom domain, not only the technologies in it, but also the diverse and challenging daily practice of organisations active in the communications sector. The course is provided by a team of 17 lecturers interactively sharing their knowledge about communications infrastructure in the context of overarching societal trends, service & network architecture, long term developments and new technology evolving from standardization initiatives. As a consequence of continuous innovation and societal developments, the course content is being updated each year including the choice of a central theme that enhances and connects the course lectures. In 2020, the themes Internet of Things, Artificial Intelligence and 5G were highlighted, likely to be updated in 2021.	
Course Contents Continuation	EE5010 Internships or ET4399 projects in telecom related industries.	
Study Goals	Overview and understanding the Communications Sector, Services, Telecom infrastructures and Value Cases serving society	
Education Method	Online lectures, one powerpoint pitch presentation given by student teams and four off-line written assignments in teams of 2 or 3 students	
Literature and Study Materials	See information on Brightspace.tudelft.nl	
Reader	Course lectures, background information and video recordings available on ET4034 Brightspace	
Prerequisites	Finalised BSc as ET4034 is meant for Master students. PhD students are also welcome and can obtain credits when successfully passing the course assignments via the Graduate School.	
Assessment	The examination of the course Together forming the basis for the examination, the ET4034 course contains five assignments of which four written documents and one opportunity to strengthen ones pitching skills presenting a self-developed Value Case to a live audience. Students are invited to work in couples or trio's on the five assignments. This information may change depending on the developments around Covid-19	
Exam Hours	Not applicable, the course assignments are four documents written offline together with a course partner. The Value Case powerpoint presentation is a pitch that takes circa 7 minutes.	
Maximum number of participants	No limitation	
Judgement	Judgement method of the course assignments The four written assignments each have a weight factor 1 and the Value Case pitch presentation has a weight factor 0,25 in the end mark of the ET4034 course.	
Self test	Not applicable	
maximum aantal deelnemers	No limitation	

IN4341	Performance Analysis	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	0/2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	This course applies probability theory and the theory of stochastic processes to the design and performance evaluation of complex networks such as man-made networks as telecommunication, computer and embedded networks and biological networks. The computation with random variables is reviewed. Markov processes and queuing theory will be introduced to the current important concept of "Quality of Service (QoS)" provisioning and to the computation of the blocking probabilities in telephony (both fixed as mobile). Several applications (e.g. the robustness of networks, epidemics in networks, the Internet shortest path routing) are also included. More details are found on brightspace.	
Study Goals	The course intends to provide students with mathematical techniques, in particular probabilistic methods and graph theory, to compare the performance of different network designs and protocols.	
Education Method	Lectures and homework after each class	
Literature and Study Materials	We follow the book Performance Analysis of Complex Networks and Systems, by P. Van Mieghem, Cambridge University Press (2014). See http://www.nas.ewi.tudelft.nl/people/Piet/bookPA.html	
Assessment	Written and closed book. A formularium is provided that can be consulted at the examination. disclaimer: information may change depending on the developments around the coronavirus.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Software Engineering 2021	
Introduction 1	The Software Engineering Research Group aims at developing a deep understanding of how people build and evolve software systems and, with that knowledge, develops novel methods, techniques and tools that advance the way in which software is built and modified. This means that we study how software is developed (e.g., by mining collaborative platforms like GitHub or setting up experiments with developers) in areas such as software testing, code reviewing, end-user development or continuous integration. At the same time, we also build tools and prototypes that alleviate some of the pains that we observe (tools like TestNForce, Cloneboard, etc.). In addition, we are also working in the area of software language design and engineering, which aims to effectively design, implement, and apply domain-specific software languages. Main areas that we are working in here are: the automatic derivation of efficient, scalable, incremental compilers and effective IDEs from high-level, declarative language definitions. At the same time, we investigate the systematic design of DSLs with an optimal tradeoff between expressivity, completeness, portability, coverage, and maintainability. A good example of a recent master thesis is Alex Nederlofs thesis Analyzing Web Applications: An Empirical Study in which he fully automatically tests 3,422 randomly selected web sites for errors and accessibility standard violations. This thesis can be found at the groups website: www.se.ewi.tudelft.nl. This website also contains information about possible master thesis projects.

CS4295	Release Engineering for Machine Learning Applications	5
Responsible Instructor	L. Miranda da Cruz	
Responsible Instructor	Dr.ing. S. Proksch	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	The world of Software Engineering has been revolutionized in the last decade. Instead of releasing software updates yearly, companies can now release multiple times per week, sometimes even per day, to their customers. This allows much quicker reactions to market demands, software failures, and is crucial to increase the business value of software. These improvements have been mostly enabled by advances in release engineering and, in this course, we will learn about the techniques and technologies that build the foundation for modern release engineering. We will go on a journey that starts at continuous integration and then moves on to continuous delivery, continuous deployment, and continuous experimentation. We will discuss the theory and the current research on various related subjects like containerization, testing, or monitoring and will put the learned theory into practice. As a running example, we will build a pipeline for a machine learning application, which -compared to traditional release engineering- poses additional challenges, like data versioning or model deployment.	
Study Goals	After following this course, students are able to... - Apply standard techniques of release engineering - Apply version control techniques to machine learning artifacts, like data or models - Design a deployment pipeline for a machine learning application - Implement quality control techniques in a machine learning pipeline - Analyze and improve existing deployment pipelines - Evaluate and document design decisions in deployment pipelines	
Education Method	- Following interactive lectures - Active participation in tutorial sessions - Reading scientific papers and gray literature - Performing a small literature survey - Implementation of a pre-defined release engineering pipeline - Deriving and implementing an improvement for the pipeline - Documenting the improvement in a scientific essay	
Assessment	Formative Assessment: - Individual group meeting for feedback on current pipeline and pipeline extension proposal - Written feedback on Table of Contents and Introduction of written essay - Individual group meeting for feedback on project progress - Written feedback on methodology and pipeline of written essay Summative Assessment: 35% Final release engineering pipeline (focus: how well is the project executed) 60% Essay (focus: how well have design decisions been evaluated and documented) 5% Presentation (focus: clarification and fraud prevention) Please note: - The different parts of the summative assessment represent grading components and need ALL to be passed to receive a positive overall grade. - There is NO resit opportunity for this course. - Partial grades are not carried over to the next academic year.	
Special Information	The course information presented in the study guide may change depending on the developments around the coronavirus.	
Co-Instructor	Prof.dr. A.E. Zaidman	

CS4405	Analysis of Concurrent and Distributed Programs	5
Responsible Instructor	Dr. B. Özkan	
Responsible Instructor	Dr. S.S. Chakraborty	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software systems are becoming highly concurrent and distributed to utilize modern multicore architectures and increasing speed and bandwidth in networks. Shared-memory concurrency in multicore programs and message-passing concurrency in distributed programs share many common abstractions and problems. In the multicore era, all performance-critical software employs some form of concurrent programming; typically shared memory concurrency. In this setting, programmers use a number of primitives to develop efficient and correct concurrent programs. To do so the programmers have to understand the behaviors of the primitives and reason about them. It is also important to match the programming paradigms and underlying architectures. For instance, traditionally programmers have assumed that a multithreaded program executed simply by interleaving the executions of its threadsa model known as sequential consistency (SC). This assumption is, however, invalidated both by mainstream multicore architectures, which often execute instructions out of order, and by compilers, whose optimizations affect the outcomes of concurrent programs. As a result, concurrent programs have more outcomes than SC allows. In the distributed setting, the units of concurrency are independent processes that do not share memory but communicate by exchanging asynchronous messages. The execution of such a system involves two main sources of nondeterminism: concurrency and partial failures. As the processes run concurrently, the exchanged messages can be delivered and processed in many different orderings. The distributed set of processes is also prone to network of process failures. The trade-off between the systems availability in the existence of failures and the consistency between the processes gives rise to a spectrum of weak consistency notions. It is important to reason about concurrency, possible failures, and consistency guarantees to implement distributed programs correctly and understand their behavior. This course aims to explore analysis techniques for concurrent and distributed programs. Outline of Lectures: Shared memory concurrency: - Abstractions for shared memory concurrency - Relaxed memory concurrency - Correctness of concurrent programs Distributed concurrency: - Distributed system components, models and assumptions - Fundamental abstractions for distributed systems	
Study Goals	This course aims to give students a deep understanding of concurrency and distribution in modern systems and hands-on experience for analyzing these systems. At the end of the course, the students will be able to: - Analyze and reason about concurrent and distributed programs - Apply and analyze existing techniques on unseen problems - Be able to pursue independent further research in the area	
Education Method	The course consists of the following education methods: - Lectures for reviewing concurrency and distribution concepts - Homeworks/assignments - Developing a course project, writing a report, and presenting it (course project) To finish the course, students (in teams) will have to: - Study several papers which will be discussed during the lectures - Deliver their assignments - Deliver and present their implementation project	
Assessment	The final grade is composed of: research project implementation) (40%) + research project report (20%) + research project presentation (20%) + homework assignment (10% + 10%). No written exam. Resits are not offered.	

CS4415	Sustainable Software Engineering	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	L. Miranda da Cruz	
Contact Hours / Week x/x/x/x	0/0/6/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	Sustainable Software Engineering is an overarching discipline that addresses the long-term consequences of designing, building, and releasing a software project. By definition, sustainability covers five main perspectives: environmental, social, individual, economic, technical. This course mainly focuses on the first, also known as Green Software Engineering. Incidentally, we will also cover some fundamental aspects of social and individual sustainability of software projects. Software Engineering (SE) has long addressed sustainability by narrowing it down to economic and technical sustainability. However, our society is facing major sustainability challenges that can no longer be overlooked by software engineers and computer scientists. It was estimated that, by 2040, the ICT sector will contribute to 14% of the global carbon footprint. Hence, environmental, social, and individual ought to be part of the equation when it comes to design, build, and release software systems. The problem is far from simple, but we need expert computer scientists to bring sustainability into the core values of the next generation of tech-leading organizations.	
Study Goals	After attending this course, you will be able to: LO1. Measure software energy consumption. LO2. Automate carbon-awareness in software development. LO3. Discuss sustainability principles. LO4. Solve sustainability issues in real software projects. LO5. Propose innovative strategies to monitor software sustainability.	
Education Method	To meet these objectives, you will be involved in a broad set of learning activities: lectures, paper reading, software analysis, software development, essay writing and presentation. These heterogenous set of activities aims at building a strong set of hard skills for energy-efficient code development combined with a strong set of soft-skills and critical thinking. Ideally, you will work on projects that will also help real-world software projects embrace a green software culture.	
Assessment	The assessment will be performed as part of the group project. It will include several steering meetings, an essay, a software repository, and a final presentation.	

IN4185	Globally Distributed Software Engineering	5
Responsible Instructor	Prof.dr.ir. D.M. van Solingen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	Exam by appointment	
Course Language	English	
Expected prior knowledge	Software Engineering (= IN2705)	
Course Contents	The course Globally Distributed Software Engineering (GDSE) will address pro's and con's of GDSE, practical consequences of GDSE, technological (in)feasabilities for GDSE, and practical experiences and examples of GDSE for example in outsourcing, off-shoring, near-shoring and multi-partner systems development. The central theme of this course is the fact that software engineering is carried out in practice more and more in globally distributed settings. This has advantages and disadvantages that need to be addressed in a practical matter when carrying such projects. The course is run asynchronous in BrightSpace. Lectures and excercises are followed digitally in weekly modules that need to be followed prior to the weekly synchronous lecture/virtual meeting. The course hours in the calendar are used for interaction with the professor and more detailed discussion and feedback. The course builds upon individual discipline in preparing the weekly modules online, in combination with group assignments during these weeks as well. Also the group assignments (in groups of 3 or 4 students) can be done virtually.	
Study Goals	The course Globally Distributed Software Engineering (GDSE) aims at teaching participants (1) the technical and organisational setting of carrying out software engineering in practice when distributed over the world, and (2) understanding best-practices in collaboration in software engineering project teams that carry out their work in a distributed setting.	
Education Method	Digital lectures, quizes, group assignments and online discussion. These are used as preparatory work prior to the weekly synchronous lectures (that are merely virtual as well), weekly group home work assignments and individual assignments.	
Computer Use	The course does not contain programming excercies. Though in the group assignment students will have to create a deliverable of choice. This can be very broad from creating a YouTube instruction video to writing an online book, or from creating a Wikipedia page to setting up tooling environment.	
Literature and Study Materials	Presentation handouts	
Assessment	Written report on lab work and literature research, individual f2f examination meeting of 30 minutes with professor. The course grade is calculated from the group assignment (25%), individual essay (25%), personal interview on GDSE course and individual essay (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Please enroll. If enrolled please pay attention that Module 1 of this course needs to be finished PRIOR to the first lecture meeting! Every week a new module is released in BrightSpace that needs to be worked through prior to the weekly synchronous meeting.	
Special Information	Please contact d.m.vansolingen@tudelft.nl	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4334	Analytics and Machine Learning for Software Engineering	5
Responsible Instructor	M. Finavaro Aniche	
Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	- Experience with programming is required - Experience with research methods is nice to have - Experience with statistics / machine / deep learning is nice to have	
Course Contents	Software repositories archive valuable software engineering data, such as source code, execution traces, historical code changes, mailing lists, and bug reports. This data contains a wealth of information about a projects status and history. Doing data science on software repositories, researchers can gain an empirically-based understanding of software development practices, and practitioners can better manage, maintain, and evolve complex software projects. Moreover, the advances in Machine Learning and AI technologies, as demonstrated by the successful application of Deep Neural Networks in various domains did not go unnoticed in the field of Software Engineering; researchers have applied machine learning to tackle different software engineering tasks. In this seminar, we will explore different software analytics tools and techniques to investigate different software engineering phenomena, and machine learning / deep learning models that tackle software engineering problems.	
Study Goals	IN4334 is a seminar course that aims to give students a deep understanding of and hands-on approach to software analytics, empirical software engineering research methods, and machine learning for software engineering. At the end of the course, the students will be able to: - Understand current literature in the area of software analytics and machine learning for software engineering - Apply software analytics techniques to extract actionable software engineering insights - Apply machine learning / deep learning algorithms to solve software engineering tasks	
Education Method	The course is a seminar, which means that we will be studying the literature in the area of software analytics and machine learning for software engineering. The course consists of the following education methods: - Self-study and presentation of papers - Development of software analytics tools and machine learning models to solve known / new problems To finish the course, students (in groups) will have to: - Study several papers (at least 10), which will be discussed during the lectures - Prepare and lead the discussion for 1 paper - Replicate existing work / propose new work The course may also feature guest lectures from top researchers in the area.	
Literature and Study Materials	Research papers are the main literature used in this course. We will share the resources with the students once the course starts.	
Prerequisites	Those are not strict prerequisites, but if you have followed those courses you will be better prepared for the Machine Learning for Software Engineering part of the course * CS4240 - Deep Learning * IN4325 - Information Retrieval	
Assessment	The final grade consists of the following items: * 80% - Results in a form of paper * 20% - Presentation of results The course does not have an exam and there will be no resit of any of the items above.	
Enrolment / Application	Each student who wants to take part in this course is *required* to: - register/enrol on Brightspace before the start of the course - participate in the first lecture of the course Failure to comply with these requirements may lead the student to be not allowed to take part in the course.	
Special Information	Due to resource constraints, this run of the course will only accept 20 students. If the number of enrolments is higher than 20, the selection procedure will be based on the student's motivation, as indicated by a short (1-2 paragraphs) motivation message that explains why the student would like to participate in this course.	
Tags	Project Research Methods Software Software Engineering	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Web Information Systems 2021	
Introduction 1	The WIS research group concentrates its research on engineering and science of the Web. The research specifically considers the role of Web data in the engineering of Web-based information systems. The group's research is aimed at improving the understanding of people's actions, interests, motivations, and behaviors on the Web, and subsequently leveraging that knowledge to build Web applications that are semantic, personalized and adaptive. This includes topics in harvesting, integrating, transforming, analyzing, and retrieving Web data, with focus on the special properties of Web data. A large portion of Web data is human-made, e.g. in social networks or Twitter streams, and this brings scientific challenges in how to effectively attribute meaning to Web data. The size of the Web brings challenges in how to efficiently store, index, and analyze data at Web scale. WIS researchers and students strive to advance the state-of-the-art in relevant disciplines like user modeling, Web science, information retrieval, Web engineering, Web data management, and crowdsourcing. A good example of a recent thesis is Catalin Stanculescus thesis with the title "Driving engagement and online social behavior of employees in an enterprise environment". This thesis contributes to the studies on enterprise gamification with a study performed at a large multinational enterprise. The student designed and implemented a modular and extensible framework for studying gamification and instantiated it as a Q&A game combined with news sharing and social connections capabilities. The framework was used to test the effectiveness of several game mechanics for promoting several types of desirable behavior concerning news sharing and knowledge acquisition.

CS4145	Crowd Computing	5
Responsible Instructor	Prof.dr.ir. A. Bozzon	
Responsible Instructor	U.K. Gadiraju	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Basic knowledge of artificial intelligence and/or human computer interaction is advised. Proficiency in at least one programming language.	

Course Contents

Crowd Computing is an emerging field that sits at the intersection of computer science and data science. Crowd computing studies how large groups of people can solve complex tasks that are currently beyond the capabilities of artificial intelligence algorithms, and that cannot be solved by a single person alone.
It involves the algorithmic engagement and coordination of people by means of Web-enabled platforms. These complex tasks are mainly focused on the creation, enrichment, and interpretation of data, making crowd computing a building block of data science. Examples of such tasks include the coordinated creation of data about real world events when electronic sensors are not available; the annotation of existing data sets to create ground truth data for the training of machine learning algorithms; and the analysis and interpretation of Web data to spot identify inappropriate content (e.g.,hate speech, or fake news).
Crowd computing is an essential tool for any data-driven company: from Facebook to Microsoft, from Google to IBM, from Spotify to Pandora, all major companies employ crowd computing to fulfil their data needs, both by involving employees, and by reaching out to anonymous crowds through online marketplaces like Amazon Mechanical Turk or Appen. Crowd computing methods therefore play an important role in the design, development and evaluation of a variety of products, services, and systems in a variety of domains.

The objective of the Crowd Computing course is to introduce the scientific and technical underpinnings of crowd computing, and to investigate how it can be used for computer science applications (e.g., information retrieval, machine learning, next-generation interfaces, and data mining) and for real world applications (e.g., cultural heritage preservation, online knowledge creation, smart cities, etc.)

The course is designed around one key challenge, the creation and consumption of (high quality) data, and will be organized around three themes:
1) Establishing data needs;
2) Fulfilling data needs with crowd computing; and
3) Evaluating the quality of the retrieved data with respect to the original data need.

Covered topics include:

1) Establishing Data Needs:
 - Requirement Elicitation
 - Requirement Analysis
 - User Modelling Properties

2) Fulfilling Data Needs with Crowd Computation:
 - Systems for/with collective intelligence (e.g., recommendation, semiautonomous systems, citizen science, crowdsourcing, and human computation systems)
 - Multi-modal Interaction (e.g., conversational systems)
 - Human Computation (e.g., worker modelling, task modelling, incentives, task assignment, recruitment)
 - Games with a purpose
 - Algorithms for Crowd Computing
 - Computational Methods for User Modelling
 - Interfaces for Crowd Computing Systems

3) Evaluating Retrieved Data:
 - Expert Evaluation
 - User Evaluation
 - Explanation of the output of Crowd Computing Systems

4) Study of Application Domains

When applicable, the course will also feature invited lectures from selected academics and professionals in the field. Since instructors of this course are also directing the Design@Scale Delft AI lab, students of this course will have the opportunity to engage with cutting-edge research projects relevant to this lab.

This Crowd Computing course is an elective for students following the Data Science and Technology Track and the Software Technology Track.
It adds to the master education offer by addressing topics that are complementary to courses like IN4325 Information Retrieval, IN4252 Web Science & Engineering, CS4065 Multimedia Search
and Recommendation, and IN4010 Artificial Intelligence Techniques.

Study Goals

After this course, students will be able to:
- Identify the requirements for a Crowd Computing system [LO1]
- Design and develop Crowd Computing systems. Support and defend the relevance and correctness of his/her choices [LO2]
- Describe and compare several Crowd Computing techniques. [LO3]
- Describe and compare design decisions in the context of Crowd Computing interaction paradigms [LO4]
- Determine which Crowd Computing technique(s) is most appropriate for being used in a certain problem domain [LO5]
- Apply the appropriate Crowd Computing technique to an application domain and evaluate the obtained results. [LO6]
- Analyse the performance of a Crowd Computing system by applying the proper evaluation measures. [LO7]

Education Method

** NB: study guide information may change depending on the developments around the coronavirus.

This course consists of 16 2-hour lectures.
Each week, a 30-minute assignment tests the knowledge acquired on the discussed topics.

Starting from Week 1, students form groups and work on a project, to be presented in week 9. Students are expected to work 6 hours per week (each) on the project assignment.

Expected workload is 32 hours for attending lectures, 24 hours of reading study material and preparing lectures, 55 hours for weekly assignments and group assignment, 24 hours for preparing final survey, and 5 hours for exam and plenary presentations (total 140 hours).

Literature and Study

Books:

Materials	- Human Computation. Author(s): Edith Law and Luis von Ahn. Synthesis Lectures on Artificial Intelligence and Machine Learning, June 2011, Vol. 5, No. 3. http://www.morganclaypool.com/doi/abs/10.2200/S00371ED1V01Y201107AIM013 - A. Marcus and A. Parameswaran. Crowdsourced Data Management: Industry and Academic Perspectives. Foundations and TrendsR in Databases, vol. 6, no. 1-2, pp. 1161, 2013. DOI: 10.1561/1900000044. https://people.eecs.berkeley.edu/~adityagp/papers/crowd-book.pdf - An Introduction to Hybrid Human-Machine Information Systems. Demartini, G., Difallah, D.E., Gadiraju, U. and Catasta, M., 2017. Foundations and Trends in Web Science, 7(1), pp.1-87. https://edu.nl/np4th Slides: available on Brightspace Articles: available on Brightspace Recommended reading: - Interaction Design: Beyond Human-Computer Interaction (4th Ed, 2015). Authors: Jenny Preece, Helen Sharp, Yvonne Rogers
Assessment	The final grade consists of the following parts: - Weekly Individual assignment, weighting 15% of the final grade - Group assignment, weighting 55% of the final grade - Final Individual Assignment (Survey), weighting 30% of the final grade The group assignment is performed collectively, but graded individually. Assignments have no re-sit opportunities. Disclaimer: information may change depending on the developments around the coronavirus.
Tags	Algorithmics Artificial intelligence Design Programming Software
Co-Instructor	J. Yang

CS4225	**Educational Technologies**	**5**
Responsible Instructor	Prof. M.M. Specht	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Course Contents	* Theories of Human Information Processing and Learning * Learning Management Systems * Learning Analytics * Personalisation and Adaptive Educational Systems * Mobile and Seamless Learning Technologies * Artificial Intelligence in Education * Realtime Learning Technologies * Project Design * Project Implementation	
Study Goals	The course will enable you to classify, understand, design and implement the core functionalities and systems for supporting human learning processes. As well current practices implemented as also approaches for technology enhanced learning currently researched will be presented. You will learn how educational technologies provide human learning process support, implement guidance and recommendation, create personalised learning support, as also give real-time feedback and support reflection of learners. In the final project you will identify a problem, design a solution based on the presented approaches and implement your own educational technology solution.	
Education Method	Lectures, weekly assignments and quiz questions, final project	
Assessment	Weekly assignments 30%, Final project 70% disclaimer: information may change depending on the developments around the coronavirus.	

IN4252	Web Science & Engineering	5

Responsible Instructor	J. Yang
Responsible Instructor	Dr. C. Lofi
Instructor	Prof.dr.ir. G.J.P.M. Houben
Contact Hours / Week x/x/x/x	2/2/0/0
Education Period	1 2
Start Education	1
Exam Period	none
Course Language	English
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.

IN4325	Information Retrieval	5

Responsible Instructor Dr. C. Hauff

Responsible Instructor J. Yang

Contact Hours / Week x/x/x/x 0/0/4/0

Education Period 3

Start Education 3

Exam Period 3

Course Language English

Expected prior knowledge

Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.

Course Contents

Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs.

Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines.

Covered topics include:

= Information Retrieval Models;
= Indexing Techniques;
= Web Search;
= Information Seeking Paradigms;
= Evaluation of information retrieval systems;
= Components of Natural Language Processing;
= Word Embedding Techniques;
= Neural Language Models;
= Deep Contextual Embeddings;
= Neural Information Retrieval

Study Goals

At the completion of this course, students will be able to:

= Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1]

= Describe and implement different indexing techniques. [Learning Objective 2]

= Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3]

= Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4]

= Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5]

= Describe typical NLP tasks and components of an NLP system. [Learning Objective 6]

= Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7]

= Compare word embedding and contextual embedding techniques. [Learning Objective 8]

= Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]

Education Method

Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments.

Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.

Literature and Study Materials

Scientific papers, course slides, course books - all resources are available on Brightspace.

Books

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press.

Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson.

Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing.

Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

Assessment

Weekly individual assignment, weighting 10% of the final grade.
Two group projects for IR and NLP, each weighting 45% of the final grade.

The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities.

disclaimer: information may change depending on the developments around the coronavirus.

IN4326	Seminar Web Information Systems	5
Responsible Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Standard bachelor-level computer science or equivalent.	
Course Contents	In this course we discuss recent developments in the area of web information systems. We select topics to discuss from the areas of: - web technology (e.g. web engineering, hypertext, adaptive web), - web data management (e.g. web data interoperability, system and data integration), - web data and semantics (e.g. ontologies, semantic web, metadata), - web data analytics (e.g. user modeling, web personalization, web information filtering and retrieval), - social web (e.g. social web data analytics, social networking, human computing), - web science (e.g. crowdsourcing, trust, data science). We discuss this content while learning about the role of scientific communication and about the scientific methodologies and approaches for conducting research in the area. In this seminar the students will have to prepare and give scientific presentations on the basis of research papers about selected topics - the topics are selected in the first session together with the students. The students will also have to attend the presentations and participate in discussions on the presented papers. In addition, students will have to write a short survey about a topic in the area of web information systems of their choice.	
Study Goals	- to expose the student to current developments in research on web information systems and be aware of the methodologies and approaches to conduct research in the area; - to familiarise the student with reading, presenting and discussing scientific literature in the area and be aware of the most important academic journals and conferences in the area (and their review processes); - to help the student in reading and writing scientific papers and choosing a topic for her/his thesis in the area.	
Education Method	Student seminar.	
Literature and Study Materials	Is provided in the seminar, depending on the chosen subjects.	
Assessment	- Quality of presentation of the scientific paper studied (15%). - Participation in the seminar discussions (10%). - Quality of paper written (75%). disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace beforehand. Students are also asked to be present and active in the first seminar session, to facilitate the proper planning of the seminar.	
Remarks	The expected workload of 5ects is distributed uniformly over the quarter. The seminar asks for active participation and therefore can only be completed as part of the first quarter edition; there is no re-sit.	
Tags	Artificial intelligence Databases	
Maximum number of participants	This course has a maximum capacity of 50 students. Students of 1) Web Information Systems group and 2) EEMCS have priority for other students.	

IN4331	Web-scale Data Management	5
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Dr. A. Katsifodimos	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Bachelor level courses in database management systems and operating systems. A prior course in distributed systems or middleware would be helpful but is not required. Programming skills are important for the final assignment.	
Course Contents	This course addresses the challenges of Data Management at Web-scale. Especially, it covers the need for large-scale distributed data storage systems. The lecture therefore introduces step-by-step increasingly complex distributed storage systems, leading up to modern implementations of different NoSQL data storage systems. The challenges arising from such systems are presented and discussed, especially focusing on the CAP theorem and the resulting trade-offs with respect to data models, transactional power, query expressivity, and replication consistency. These discussions lead to different variants of NoSQL database systems, like Key-Value Stores, Document Stores, Wide-Columnar stores, and Graph Databases. The advantages, disadvantages, and general properties of these systems are discusses in more detail. There is special focus on distributed transactions and consistency guarantees of different data management systems and methods.	
Study Goals	At the end of this course the student can - assess the nature of a given storage problem, and can select a suitable technology for solving it - understands the different data models encountered in Web Data Management, and their impact on modelling and querying - understands the issues arising from distributing and replicating data, especially with respect to the CAP theorem -understands the trade-offs which can be chosen within the design space of the CAP theorem -categorize and explain modern NoSQL databases within the framework of the previously mentioned trade-offs	
Education Method	Lectures and assignments	
Literature and Study Materials	Course slides and Lecture Videos	
Books	Literature mentioned in the lecture, mostly research papers.	
Assessment	Depending on the year, there are group assignments with final presentation, written report, and oral interviews. Final grade will be the weighted average of the three partial grades (the weights will be announced during the lecture depending on the available topics and assignments in the current quarter). The interview covers questions from the topics of the lecture. The interview can be resit, the assignment & presentation cannot. disclaimer: information may change depending on the developments around the coronavirus.	

Special Programmes 2021	
Introduction 1	There are three Special programmes: Bioinformatics (BI). Information Architecture (AI) Cyber Security

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Information Architecture 2021

Introduction 1

Compulsory Courses CS-IA:
SEN1121 Complex Systems engineering
SEN1611 I&C Architecture Design
SEN1622 I&C Services Design
IN4325 Information Retrieval
SEN1141 Managing Multi Actor Decision Making
IN4331 Web Data Management
IN4252 Web Science & Engineering

And in addition to said compulsory Information Architecture courses:

- Students of the Data Science & Technology track need to take 3 additional common core courses.
- Students of the Software Technology track need to take 4 additional common core courses.
- Students of the Artificial Technology track need to take 4 additional common core courses.

Final Year project in Information Architecture (45 EC)
IEP approved by Information Architecture Coordinator

IN4252	Web Science & Engineering	5
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4325	Information Retrieval	5
Responsible Instructor	Dr. C. Hauff	
Responsible Instructor	J. Yang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.	
Course Contents	Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs. Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines. Covered topics include: = Information Retrieval Models; = Indexing Techniques; = Web Search; = Information Seeking Paradigms; = Evaluation of information retrieval systems; = Components of Natural Language Processing; = Word Embedding Techniques; = Neural Language Models; = Deep Contextual Embeddings; = Neural Information Retrieval	
Study Goals	At the completion of this course, students will be able to: = Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1] = Describe and implement different indexing techniques. [Learning Objective 2] = Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3] = Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4] = Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5] = Describe typical NLP tasks and components of an NLP system. [Learning Objective 6] = Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7] = Compare word embedding and contextual embedding techniques. [Learning Objective 8] = Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]	
Education Method	Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments. Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.	
Literature and Study Materials	Scientific papers, course slides, course books - all resources are available on Brightspace.	
Books	Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA. Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press. Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson. Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing. Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".	
Assessment	Weekly individual assignment, weighting 10% of the final grade. Two group projects for IR and NLP, each weighting 45% of the final grade. The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities. disclaimer: information may change depending on the developments around the coronavirus.	

IN4331	Web-scale Data Management	5
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Dr. A. Katsifodimos	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Bachelor level courses in database management systems and operating systems. A prior course in distributed systems or middleware would be helpful but is not required. Programming skills are important for the final assignment.	
Course Contents	This course addresses the challenges of Data Management at Web-scale. Especially, it covers the need for large-scale distributed data storage systems. The lecture therefore introduces step-by-step increasingly complex distributed storage systems, leading up to modern implementations of different NoSQL data storage systems. The challenges arising from such systems are presented and discussed, especially focusing on the CAP theorem and the resulting trade-offs with respect to data models, transactional power, query expressivity, and replication consistency. These discussions lead to different variants of NoSQL database systems, like Key-Value Stores, Document Stores, Wide-Columnar stores, and Graph Databases. The advantages, disadvantages, and general properties of these systems are discusses in more detail. There is special focus on distributed transactions and consistency guarantees of different data management systems and methods.	
Study Goals	At the end of this course the student can - assess the nature of a given storage problem, and can select a suitable technology for solving it - understands the different data models encountered in Web Data Management, and their impact on modelling and querying - understands the issues arising from distributing and replicating data, especially with respect to the CAP theorem -understands the trade-offs which can be chosen within the design space of the CAP theorem -categorize and explain modern NoSQL databases within the framework of the previously mentioned trade-offs	
Education Method	Lectures and assignments	
Literature and Study Materials	Course slides and Lecture Videos	
Books	Literature mentioned in the lecture, mostly research papers.	
Assessment	Depending on the year, there are group assignments with final presentation, written report, and oral interviews. Final grade will be the weighted average of the three partial grades (the weights will be announced during the lecture depending on the available topics and assignments in the current quarter). The interview covers questions from the topics of the lecture. The interview can be resit, the assignment & presentation cannot. disclaimer: information may change depending on the developments around the coronavirus.	

SEN1121	Complex Systems Engineering	5
Module Manager	Prof.dr. F.M. Brazier	
Instructor	Y. Huang	
Instructor	Prof.dr. F.M. Brazier	
Instructor	Dr. P.H.G. van Langen	
Responsible for assignments	Prof.dr. F.M. Brazier	
Co-responsible for assignments	Y. Huang	
Contact Hours / Week x/x/x/x	4/0/0/0 and interactive lab sessions	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	The course introduces design thinking in socio-technical systems as the core orientation of the entire SEPAM master. Students learn about designing complex technological systems in multi-actor environments. It follows a storyline from understanding socio-technical systems towards creating artefacts to realize change in socio-technical systems. Different perspectives on systems design are discussed to provide students with a background for working with designers from different disciplines. Thereby, the course lays the foundation for further design-oriented courses. Typical questions the module will address are: - How to explore a problem situation? - How to oversee a design challenge? - How to formulate a design task? - What is a suitable design approach for specific design challenges?	
Study Goals	After completion of the course, the student is able to - use and differentiate discuss concepts and terminology related to the design in socio-technical systems - analyze a problem situation in complex, technological, large-scale systems in multi-actor environments - use and combine methods and tools for requirements analysis and conceptual design - use and compare methods and tools that facilitate systems design and engineering - apply and compare methods and tools related to decision, project and risk management - determine a suitable design approach for specific socio-technical design challenges - use and combine methods and tools for systems analysis and modelling to address complexity and uncertainty in design processes	
Education Method	Lectures, labs, assignments	
Literature and Study Materials	The main content of this course will be included in a (new) syllabus that will be made available in BrightSpace. The concepts and topics discussed in the course will be supported by regular reading of papers, articles and book chapters as appropriate. These reading materials will also be available in Brightspace.	
Assessment	The overall grade for this course consists of 2 sub grades: 1) Exam: The exam will check the theoretical knowledge as well as the ability to apply and reflect on the theory taught. The digital exam is open book and contributes to 70% of the final grade. In case of unforeseen circumstances or measures resulting from COVID-19, the digital exam will be online. 2) Assignments: During the exercises, theory from the lectures will be applied. The exercises contribute to 30% of the final grade. Both parts must be completed with a passing grade (at least 5.75) in order to pass the course.	
Elective	Yes	
Tags	Analysis Design Group Dynamics/Project Organisation Modelling Projects	

SEN1141	Managing Multi-actor Decision-making	5
Module Manager	Dr. M.L.C. de Bruijne	
Instructor	B. Wagner	
Instructor	Dr.ir. E. Minkman	
Instructor	F. Hirschhorn Zonana	
Instructor	Dr. F.S. Gürses	
Instructor	Prof.mr.dr. J.A. de Bruijn	
Instructor	Dr. M. Leijten	
Instructor	Ir. B.M. Steenhuisen	
Responsible for assignments	Dr. M.L.C. de Bruijne	
Co-responsible for assignments	Ir. B.M. Steenhuisen	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	Previous courses have focused on, first, diagnosing the complexity of systems and problems and, then, designing for systems and their institutions in the midst of this complexity. This course on multi-actor decision-making goes one step further: (reflecting on) realizing change. We focus on the actors in the system. How do they behave, individually and collectively? What does the system look like from an actor perspective? Why does intentional change often appear so hard? What limits the designability of systems? By studying characteristics of complex engineering systems (e.g. the increased importance of IT for design, the increased interdependencies between systems) as well as the specific actor characteristics (e.g. the global character of the manufacturers and supply chains, issues of plurality and inclusiveness we identify key factors that affect intentional change (i.e. the wickedness of problems, public values, strategic behaviour of actors and dynamics of the system). How do these factors affect decision-making and design and change of complex engineering systems? How can we assess the consequences of this decision-making complexity and how does this affect the potential for interventions for (re)design in these systems in deliberate and responsible ways? In this course, we identify key characteristics which affect multi-actor decision making in complex engineering designs. Furthermore, we assess how and under which conditions process managerial tactics may help to manage multi-actor decision-making with its plural rationalities and dynamics of negotiation and design.	
Study Goals	At the end of this course, students will be able to explain why actors in networks behave as they do, and how their behaviour evolves describe different network structures and their practical implications assess the designability of institutions, systems, policies and technologies argue why certain process managerial tactics might work under which conditions create case-specific process managerial tactics to anticipate complex decision-making. assess the quality of process designs reflect on multi-actor decision-making in light of broader scientific and public policy trends and debates	
Education Method	Lectures Assignments Readings	
Assessment	Assessments group assignment analysing a multi-actor decision making process (1/2) individual assignments which enable students to reflect on prescribed literature (1/2)	

SEN1611	I&C Architecture Design	5
Module Manager	Prof.dr.ir. M.F.W.H.A. Janssen	
Instructor	Prof.dr.ir. N. Bharosa	
Responsible for assignments	Prof.dr.ir. M.F.W.H.A. Janssen	
Co-responsible for assignments	G.A. De Reuver	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	· Basic knowledge of databases, including Entity-Relationship Diagram (ERD) · Basic programming skills (such as JavaScript) · Basic knowledge of Unified Modelling Language (UML) and software engineering (agile development) · Basic understanding of multi-actor system and stakeholder analyses methods This prior knowledge are part of the linkage program.	
Course Contents	More and more data is available collected by social media or the Internet of Things (IoT) which is processed by a variety of systems. The ability to execute and process data is a key capability required by public and private organizations. The large variety of heterogeneous data, applications and business processes results in a complex and fragmented landscape that need to be dealt with when architecting and engineering new systems. Path dependencies and legacy block the easy integration, whereas at the same time new technologies appear that need to be integrated in the complex systems landscape. The purpose of this course is to teach the architectural design of innovative and large-scale ICT infrastructures and services in the light of the challenges imposed by the requirements from the systems physical, economic and social environment. Emphasis will be put on the concepts and role of ICT-architecture and modelling in order to properly design ICT solutions within a multi-actor context. For this purpose application and data integration technologies and information and system quality theories will be addressed.	
Study Goals	-The student is familiar with the state-of-the-art knowledge of ICT-architecting, design and governance within the field of large-scale ICT-systems within a multi-actor context -The student is able to describe basic concepts related to architecting and designing large and complex ICT-infrastructures and service systems within a multi-actor context. -The student should master architecture theories, methods and tools with the ability to combine and switch between them when dealing with complex ICT-problems. -The student is able to structure and analyze problems and identify dilemmas arising during the design process with regard to designing large ICT-systems within a multi-actor context. -The student is able to apply system engineering and architecture-based approaches, methods and tools to deal with problems with regard to designing large ICT-systems. -The student is able to report about the use of architectural concepts, methods and tools for translating business needs into ICT-designs within constellation of public and private actors.	
Education Method	· Lectures. The lectures are aimed at providing an overview of the knowledge of this course. During the lectures exercises are given to practice to internalize the knowledge. · Guest lectures (obliged). The guest lectures are aimed at giving an overview of the state-of-the art in practice and provide insight into practical challenges when using ICT in a multi-actor domain. · Exercises During some of the lectures assignments are given to practice what is learned. · Assignments. Various assignments are given which together create the final report. This includes developing various architectural models. · Presentation. Students should present their results during a lectures. · Report. The results should be reported in a concise report including the argumentation of design choices, the resulting architecture and an evaluation	
Computer Use	Architecture modelling in Archimate tool (http://www.archimatetool.com/) BPMN modelling using Eunomia process builder (http://www.eunomia-process.com/), Bizagi (http://www.bizagi.com/) or LucidChart (https://www.lucidchart.com/)	
Literature and Study Materials	slides book (open access e-book) Nitesh Bharosa, Remco van Wijk, Niels de Winne, Marijn F.W.H.A. Janssen (2015). Challenging the chain. Governing the Automated Exchange and Processing of Business Information. ISBN: 978-1-61499-497-8 (open access). This book covers a range from governance to technical issues and integrates them. Open access book: http://ebooks.iospress.nl/book/challenging-the-chain-governing-the-automated-exchange-and-processing-of-business-information papers Ross Jeanne W. (2003). Creating a Strategic IT Architecture Competency: Learning in Stages, MISQ Executive, 2(1), 31-43. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Written by the Institute of Electrical and Electronic Engineers (IEEE Std.1471-2000, 2006) Marc Lankhorst and the ArchiMate team (2004). ArchiMate Language Primer Introduction to the ArchiMate Modelling Language for Enterprise Architecture. Telematica Instituut/Archimate Consortium, August 2004) Stephen A. White (2005). Introduction to BPMN. IBM Corporation, February 2005 Janssen, Marijn (2009). Framing Enterprise Architecture: A meta-framework for analyzing architectural efforts in organizations. In: Gary Doucet, John Gøtze, Pallab Saha and Scott Bernard (eds.) Coherency Management: Architecting the Enterprise for Alignment, Agility, and Assurance. International Enterprise Architecture Institute (ISBN: 978-1438996073 978-1438996073). Pp. 99-120. Bharosa, N. & Marijn Janssen (2015). Principle-based design: A methodology and principles for capitalizing design experiences for information quality assurance. Journal of Homeland Security and Emergency Management (DGJHSEM), Vol. 12, no. 3, pp. 469-496, DOI: 10.1515/jhsem-2014-0073 Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. ACM computing surveys (CSUR), vol. 41, no. 3, article 16. Peterson, R. (2004). Crafting information technology governance. Information Systems Management, vol. 21, no. 4, pp. 7-22. Weill, P., & Ross, J. (2005). Designing IT governance. MIT Sloan Management Review, vol. 46, no. 2, pp. 26-34.	
Assessment	Group assignment - The assignment is assessed by a report which should adhere to the APA reference standards (60%) Exam (40%) Each grade should be 5.75 or higher	
Elective	Yes	
Tags	Information & Communication	

	Modelling Prototyping Technology
Targetgroup	SEN I&C track Information Architecture (IA) students Electives for MOT, EPA, Computer science students, and non-TBM students having the required knowledge in this field are welcome

SEN1622	**I&C Service Design**	**5**
Module Manager	Dr. Y. Ding	
Instructor	Dr. Y. Ding	
Instructor	Prof.dr.ir. G.A. de Reuver	
Responsible for assignments	Dr. Y. Ding	
Co-responsible for assignments	Prof.dr.ir. G.A. de Reuver	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	Mobile apps, Internet-of-things, cloud architectures and sensors are enabling a range of new I&C services. These services are typically offered in ecosystems of interdependent actors. However, designing services that add value for users as well as ecosystem stakeholders is challenging. In this course, you learn how to design practical ICT services and develop a valorization plan. The course contains two main parts. In the first part, it is all about designing innovative services that meet the needs of users. You will design service mockups and interview users to test your ideas. In the second part, it is about bringing that service idea to reality. You will design a valorization plan that specifies how to deal with external technologies, stakeholders and revenues. The course is practical and hands-on: you will create, test and plan your service ideas. At the same time, all is theory-informed and you will learn how to support design choices using relevant kernel and design theories.	
Study Goals	After the course, you are able to: - Analyze practical ICT services by applying concepts of design science research, action design research, agile development and service engineering - Evaluate practical requirements and make informed choices on supporting technologies and platforms, with specific attention to autonomous driving, medical IoT, AI, XR, and edge computing technologies - Evaluate an ICT-enabled service concept through semi-structured interviews - Design and illustrate a value-adding service concept driven by ICT - Design a valorization plan that explicitly covers how and when to involve external stakeholders in a design process and integrate the ICT service with value network	
Education Method	Lectures, design project, peer review, presentations, guest lecture from industry.	
Assessment	Group essay = 3 EC; Group mockup = 1.25 EC, Individual reflection document = 0.75 EC All three marks should be >5.75. Participation in peer reviewing, guest lecture and presentations is mandatory.	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Bioinformatics 2021

Introduction 1

General Setup 2020-2021
I. DST/AIT Common Core courses (>20EC: choose 4 out of 9) or ST Common Core courses (>25EC: choose 5 out of 11)
II. BI core courses, 25EC
III. BI specialization courses, >15EC
IV. Literature study, 10EC
V. Free electives, >10EC
VI. Thesis project, 45EC

--> IEP must be approved by the Bioinformatics MSc Coordinator

I DST/AIT Common Core courses (>20EC: choose 4 out of 9) or ST Common Core courses (>25EC: choose 5 out of 11)

II Compulsory Bioinformatics courses (25EC)
1. CS4250 Selected topics in molecular biology, 5EC
2. CS4255 Algorithms for sequence-based bioinformatics, 5EC
3. CS4176 Algorithms for network-based bioinformatics, 5EC
4. CS4260 Machine learning in bioinformatics, 5EC
5. CS4329 Recent topics in bioinformatics, 5EC

CS4176 --> Old name: IN4176 Functional Genomics & Systems Biology
CS4329 --> Old name: IN4329 Advanced Bioinformatics

III Specialization courses: choose >15EC

Bioinformatics specialization Courses Q1
1. CS4070 Multivariate Data Analysis, 5EC
2. EE4C06 Networking, 5EC
3. IN4049TU Introduction to High Performance Computing, 6EC
4. IN4252 Web Science & Engineering, 5EC
5. IN4344 Advanced Algorithms, 5EC
6. IN4010(-12) Artificial Intelligence Techniques, 6EC
7. IN4307 Medical Visualization, 5EC

IN4344 --> Old course code: IN4301

Bioinformatics specialization Courses Q2
1. IN4086-14 Data Visualization, 6EC
2. IN4150 Distributed Algorithms, 6EC
3. NB4130TU Biologic, 3EC
4. CS4220 Machine Learning 1, 5EC

CS4220 --> Old course code: IN4085 Pattern Recognition

Bioinformatics specialization Courses Q3
1. CS4240 Deep Learning, 5EC
2. CS4195 Modeling and Data Analysis in Complex Networks, 5EC
3. CS4230 Machine Learning 2, 5EC
4. IN4391 Distributed Systems, 5EC
5. IN4325 Information Retrieval, 5EC
6. IN4315 Software Architecture, 5EC

CS4240 --> Old course code: CS4180 Deep Learning
CS4230 --> IN4320 Machine Learning has been replaced by CS4230 Machine Learning 2 (new name, same course content)

Bioinformatics specialization Courses Q4
1. CS4205 Evolutionary Algorithms, 5EC
2. IN4331 Web Data Management, 5EC
3. CS4290 Seminar Distributed Machine Learning Systems, 5EC
4. CS4245 Seminar Computer Vision by Deep Learning, 5EC

CS4245 --> Old code: IN4393-16 Computer Vision

IV Literature study (10EC)
IN4306 Literature Survey, 10EC

V Free Electives (10EC)

VI Thesis project (45EC) with Bioinformatics group
IN5000 Final Project, 45EC

CS4070	Multivariate Data Analysis	5

Responsible Instructor Dr.ir. F.H. van der Meulen

Responsible Instructor Dr. D.M.J. Tax

Contact Hours / Week x/x/x/x 4/4/0/0

Education Period 1
2

Start Education 1

Exam Period 1
2
3

Course Language English

Expected prior knowledge

* Introduction Probability Theory and Statistics: see for instance

A Modern Introduction to Probability and Statistics
Understanding Why and How
Series: Springer Texts in Statistics
Dekking, F.M., Kraaikamp, C., Lopuhaä, H.P., Meester, L.E.
2005, XVI, 488 p. 120 illus., Hardcover
ISBN: 1-85233-896-2

* Basic calculus

* Linear Algebra: matrix multiplication, the inverse of a matrix, the transpose of a matrix, least square solution.
see:

David C. Lay: Linear Algebra and Its Applications
ISBN-10: 0321385179 ISBN-13: 9780321385178
©2012 Pearson)

Course Contents

PART I:
Stochastic models will be developed on the basis of probability theory. Probability theory describes the behavior of certain phenomena in terms of how likely it is that certain values will occur. Central features of the models will be discussed are random variables, probability density functions, and the expected value operator. In describing random processes and signals, the correlation function and conditional probabilities play a central role.

It addresses the following subjects:
1. Random variables. Matlab exercise on estimation of PDF, expected value and variance.
2. Refresher correlation. Calculating with correlation functions.
3. Random processes, correlation function, stationarity, wide sense stationarity, estimation of correlation function (Matlab exercise).
4. Random signal processing, power spectral density function, white noise.
5. AR processes, linear prediction: theory and Matlab exercise.
6. Markov chains.

PART II:
A course in advanced statistics about linear models, Bayesian inference, classification problems, Gaussian processes and Markov Chain Monte Carlo.

Study Goals

PART I:
1. Probability Theory
- Conditional) probabilities, the law of total probability, and Bayes rule.
- Solve probability problems that require the use of axioms of probability.

2. Definition and Description of Random Variables and Processes
PDF, PMF, CDF, Covariance, Correlation- Determine if a given PDF, PMF, CDF, variance, (auto/cross-)correlation(-function), (auto/cross-)covariance(-function), power spectral density complies with (theoretical and analytical) requirements.
- Convert the description of a probabilistic problem into a probabilistic model using PDF, PMF, or CDF.

3. PDF/PMF and Expected Value
Calculate the various forms of expected value of (combinations of) random variables and random processes
- For a given (amplitude continuous/discrete and time continuous/discrete) probability model calculate the following probabilistic (marginal, joint and conditional) characterizations: PDF, PMF, CDF, probability of an event, expected value, variance, covariance, correlation, correlation coefficient, auto/crosscorrelation function, auto/crosscovariance function, (cross) power spectral density.
- Calculate the PDF, PMF, expected value and variance of a derived random variable.

4. Properties of Random Processes
- Independence, orthogonality, uncorrelated, whiteness, IID- Determine if random variables/processes have the following properties: independent, orthogonal, uncorrelated, white, Poisson, Gaussian, Bernoulli, Markov, IID, stationary, WSS, ergodic.
- Calculate the expected value, variance, auto/crosscorrelation(function), auto/crosscovariance(function), power spectral density of a linear combination of random variables and of a linearly filtered (WSS, amplitude discrete/continuous, time discrete/continuous) random process.

5. Large NumbersCentral limit theorem, law of large numbers
- Solve problems that require the use of the central limit theorem in an engineering context
- Explain the law of the large numbers in an engineering context.

6. Statistical Estimators
- Estimated mean, variance, and correlation function
- Given a set of outcomes, sample functions or realizations, calculate estimators for expected value, variance, and (auto-)correlation function.

12. Application to Engineering Problems and Simulations
- Select and translate a simple electrical engineering or computer science problem into mathematical probability model. The emphasis is on problems in signal and image processing, telecommunication, and media and knowledge technology. The class of probability models encompasses the following random variables/processes: Bernoulli, exponential, binomial, Poisson, Gaussian, uniform.
- Justify and reflect on the approach taken in calculating or simulating (MatLab) the following probabilistic properties: PDF, PMF, expected value, variance, autocorrelation function, autocovariance function.

	PART II: After finishing this course, the student is able to apply and derive statistical methods from both the frequentist and Bayesian perspective for - linear models - classification problems - clustering problems - Gaussian process regression The student is able to give a clear presentation about the underlying statistical theory. The student is able to compute several statistical characteristics by hand.
Education Method	PART I: Lectures, working groups (problem solving), laboratory work (a Matlab exercise) Workload is around 15 hours for attending lectures, 5 hours of reading study material and preparing lectures, 15 hours for the lab course, 20 hours for preparing the exam, 3 hours for the exam, and 8 hours for a final report (66 hours in total). PART II: Classes and weekly exercises.
Books	PART I: R.D. Yates and D.J. Goodman, "Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers", ISBN 0-471-17837-3, John Wiley and Sons, New York, 2005, Second Edition. PART II: Simon Rogers and Mark Girolami "A first coruse in machine learning, 2nd edition" Chapman & Hall From this book chapters 1--4, 8 and 9 will be covered.
Assessment	The final grade is the average of the grades you get for part (I) and (II). For part (I) there is a lab and written exam, where the grade is determined by the exam, and the lab assignment should be Passed. If you fail the lab assignment, you'll get a second chance to submit around the time the resit. For part (II), there will be an on-campus written exam. To pass the course, the average should be 5.8 or higher, and the grade of each individual part should be a 5.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.
Exam Hours	PART I: Online exam of 3 hours. PART 2: On campus 3 hour written exam PART II: Written exam of 3 hours.
Permitted Materials during Tests	PART I: Self made notes on a two-sided written A4 sheet. Calculator. PART II: none
Remarks	PART II: This course is particularly interesting for students that are interested in statistical exploratory and quantitative techniques to analyse multivariate data.

CS4176	Algorithms for network-based bioinformatics	5
Responsible Instructor	Prof.dr.ir. M.J.T. Reinders	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	Exam by appointment	
Course Language	English	
Expected prior knowledge	The student is expected to have a basic knowledge of molecular biology, statistics and linear algebra. It is advisable to have followed CS4220 Machine Learning 1 (old code: IN4085 Pattern Recognition).	
Course Contents	Molecular biology is concerned with the study of the presence of and interactions between molecules, at the cellular and sub-cellular level. In bioinformatics and systems biology, algorithms and tools are developed to model these interactions, with various goals: predicting yet unobserved interactions, assigning functions to yet unknown molecules through their relations with known molecules; predicting certain phenotypes such as diseases; or just to build up biological knowledge in a structured way. Such interaction models are often best modelled as networks or graphs, which opens up the possibility of using a large number of readily available algorithms for inferring networks, performing simulations of biology, optimising paths or flows through networks, graph-based data integration and graph mining. Many of these algorithms can be applied (sometimes with slight alterations) to solve a particular biological problem, such as modeling transcriptional regulation or predicting protein interaction/complex formation, but also to derive systems behaviour by breaking down networks into modules or motifs with certain characteristics. In this course, we will first give a brief overview of molecular biology, the advent of high-throughput measurement techniques and large databases containing biological knowledge, and the importance of networks to model all this. We will highlight a number of peculiar features of biological networks. Next, a number of basic network models (linear, Boolean, Bayesian) will be discussed, as well as methods of inferring these from observed measurement data. Building on the network inference methods, a number of ways of integrating various data sources and databases to refine biological networks will be discussed, with specific attention to the use of sequence information to refine transcription regulation networks. Finally, we will give some examples of algorithms exploiting the networks found to learn about biology, specifically for inspecting protein interaction networks and for finding active subnetworks.	
Study Goals	After succesfully completing this course, a student is able to: list the basic elements of a living cell and their interactions, and describe how these can be measured; explain what type of mathematical model is applicable to what measurement(s), at what level(s), in a given systems biology problem; read and comment upon recent network-based computational biology literature; discuss the state-of-the-art in systems biology and integrative bioinformatics, and future challenges	
Education Method	The course consists of a mixture of lectures by the teachers and paper presentations by one or more of the students. Each paper presentation will be followed by a in-depth discussion. There will also be a practical session allowing students to get hands-on experience with network models.	
Literature and Study Materials	Slides, collection of papers and lab course manual (Brightspace).	
Assessment	Students are required to write a proposal for a research project in which they clearly state the biological problem to be solved, the necessary data, the computational approach as well as the innovative parts of the approach. This proposal will be graded. Next, the paper presentations as well as discussions will be graded. The final grade for the course will be based on all these grades: proposal 80% and the presentations/discussions (20%). disclaimer: information may change depending on the developments around the coronavirus.	
Remarks	As students depend on each other (to present the material to the class), a commitment to follow the course through to the end is required.	

CS4195	Modeling and Data Analysis in Complex Networks	5
Responsible Instructor	H. Wang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The assignment and final project require basic programming skill.	
Course Contents	Big Data is mostly obtained from features of components and the interactions among components in large complex systems. Examples are (1) end user features and interactions in both online and real-world social networks like Twitter, LinkedIn (2) data from content sharing platforms such as YouTube (3) physiological data of the brain and (4) stock prices etc. in economic systems. Such a dataset is networked in nature i.e. the data of the system components or interactions are (cor)related to each other. This course introduces the basic methodologies to analyze, model, interpret and possibly to predict such Networked Data, combining advances from network science, modeling of dynamic processes and statistical physics, beyond machine learning algorithms. These methods will be applied to diverse real-world datasets obtained from e.g. Facebook, LinkedIn, YouTube, the brain etc.	
Study Goals	After this course, students could construct a network based on the dataset, characterize and model the network in order to e.g. detect patterns and anomalies, model the data via dynamic processes (e.g. viral spreading) on networks to decode the underlying governing mechanisms of e.g. information/error/behavior contagion and to predict e.g. the popularity of a product, news, disease, computer virus, control the contagion process such as maximize the information prevalence and market share. Students could obtain an overview of the Msc/Phd projects on the frontiers of networked data analysis.	
Education Method	In total, there will be about 7 lectures. Students will also learn via an assignment and a final project (each group gets individual supervision).	
Assessment	The final grade is based on the assignment (20%) and final project (80%). There is no resit for both the assignment and the project. disclaimer: information may change depending on the developments around the coronavirus.	

CS4205	Evolutionary Algorithms	5
Responsible Instructor	Prof.dr. P.A.N. Bosman	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Must have the ability to program your own (extensions of) evolutionary algorithms (e.g., C, C++, Java). Further, although no particular high-level prior knowledge is expected, a basic understanding of key aspects in linear algebra (e.g., matrix multiplications, inversions, decompositions, etc.), probability theory and statistics (e.g., normal distribution, statistical hypothesis testing), algorithm design (e.g., efficient local search, heuristics), complexity theory (e.g., big-O notation, algorithm analysis), will be very helpful.	
Course Contents	In this course we consider a specific subfield of Artificial Intelligence: Evolutionary Algorithms (EAs). These algorithms, sometimes also identified as being part of the class of bio-inspired algorithms, have as a metaphor the concept of natural evolution, i.e., the mechanisms by which, the fittest individuals in a population survive, reproduce, and in doing so, over time, change to be better equipped to thrive in their environment. Initiated in the 60s and 70s of the 20th century, research on EAs has progressed immensely. Today, EAs are being used to solve real-world problems in many areas, e.g. to optimize the layout of electrical wind farms, to automatically create radiation therapy treatment plans, and to optimize the architectures of deep neural networks. This course covers a spectrum of topics in EAs, ranging from basic concepts to advanced, recent, and state-of-the-art research, and ranging from theoretical to applied. In particular, topics include genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, optimal mixing evolutionary algorithms, multi-objective optimization, and real-world applications. The course is planned to have 7 lectures and 2 practical assignments. The first practical assignment pertains to experimenting with already implemented EAs on predefined problems. The second practical assignment offers more freedom, allowing you, in a group, to build your own EA (this may vary depending on student numbers and other circumstances). disclaimer: information may change depending on the developments around the coronavirus.	
Study Goals	Upon successful completion of this course, students will be able to: 1) Explain the key concepts underlying the main streams in Evolutionary Algorithm (EA) research, with in particular genetic algorithms, evolution strategies, genetic programming, estimation-of-distribution algorithms, and optimal mixing evolutionary algorithms. 2) Explain key ingredients underlying the rationale of when these algorithms work and when they do not work. In particular: schema analysis and how the match between the search bias of an EA and the fitness landscape is influenced by aspects such as variable dependencies and multi-modality. 3) Name and explain key research lines along which state-of-the-art research in EAs is done to achieve more robust, efficient, and effective EAs. 4) Identify good opportunities for using EAs, or hybrid versions thereof, in practice. 5) Properly (scientifically) experiment with EAs as well as program your own.	
Education Method	7 Lectures 2 Lab projects disclaimer: information may change depending on the developments around the coronavirus.	
Literature and Study Materials	Papers and slides that will be made available.	
Assessment	The final grade is based on 60% written exam, 40% lab practical work. disclaimer: information may change depending on the developments around the coronavirus. In particular, there may be no written exam. In that case, there will likely be several smaller practical assignments and a large practical assignment at the end of the course. Both parts will then likely count for 50%. For the large assignment there will also be chance to resit this part by means of a repair option through an oral examination of the lecture contents.	
Permitted Materials during Tests	None	
Tags	Algorithmics Artificial intelligence Optimalisation	

CS4220	Machine Learning 1	5
Responsible Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/4/0/0 Lectures + 0/2/0/0 lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	This course is required for CS4230 Machine Learning 2	
Expected prior knowledge	For the course CS4220, you should know the terminology that is taught in the course CSE2510. So, please have a look at the content of CSE2510 in Brightspace. It is not required that you followed the course CSE2510 in full, or made the exam.	
Course Contents	Recapitulation of (un)supervised learning, classification, decision theory overfitting. Complexity, regularisation, and support vector classifiers. Regression, linear and kernel regression. Bayesian learning, graphical models. Clustering and mixture models, the EM algorithm. Feature selection and extraction, PCA. Design and analysis of ML experiments.	
Study Goals	After successfully completing this course, the student is able to: recognise machine learning problems and select algorithms to solve them; read and comprehend recent articles in engineering-oriented pattern recognition journals, such as IEEE Tr. on PAMI; construct a learning system to solve a given simple machine learning problem, and able to implement algorithms from literature.	
Education Method	Lectures, laboratory work (mathematical exercises and computer exercises)	
Assessment	One final exam for 100% of the grade. This can be a written exam or an online exam, depending on the Corona situation. disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	M. Loog	

CS4230	Machine Learning 2	5
Responsible Instructor	M. Loog	
Instructor	Dr.ir. J.H. Krijthe	
Instructor	Dr. F.A. Oliehoek	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/4/4	
Education Period	3 4	
Start Education	3	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course is the more advanced and research oriented follow-up to CS4220 [Machine Learning 1]. The content of the latter is, therefore, expected as prior knowledge.	
Course Contents	The course will treat a number of machine learning theories and techniques in detail and on an advanced level. Possible topics : - learning theory - Bayesian networks - online learning - Rademacher complexity - Markov decision processes - semi-supervised learning - reinforcement learning - active learning - causal reasoning and discovery	
Study Goals	After successfully completing the course, the student is able to apply the techniques and theories that have been covered in the course. In addition, they are able to develop learning strategies for new and previously unseen situations. Moreover, the student can provide reasoned justifications for these strategies based, for instance, on theory and/or experiment.	
Education Method	Lectures + Q&A sessions	
Assessment	Grading is based on two parts. Following the lectures -- we have about 11 of those, there is an individual assignment that will be graded pass/fail. In addition, there is a written examination that will be graded on a scale from 1 to 10. You pass the course when you both have a pass for the assignment and a passing grade for the written exam. Upon passing the course, your final grade will be the grade for the exam. Finally, note that there is a resit for the written examination, but not for the report. disclaimer: information may change depending on the developments around the coronavirus.	

CS4240	Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	Dr. D.M.J. Tax	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Required for	Seminar Computer Vision by Deep Learning	
Expected prior knowledge	Basic pattern Recognition, Basic Machine learning, Basic statistics, Basic probability theory, Programming experience (python + numpy). Note: Without some affinity/experience with python (and numpy) programming it might still be possible to pass the course, but will become quite difficult.	
Course Contents	In this course we will look at a specific field of Artificial Intelligence and Machine Learning: Deep learning. Deep learning has shown remarkable success with large data sets and unstructured input data such as raw images/audio/text. Topics include: feed forward networks, back-propagation, optimization, convolutional nets, recurrent nets, self-attention, unsupervised methods. The course will have lectures, a seminar, a lab practical and a project: - The lectures will be on generic topics, following the book; building the backbone. - The lab assignments will have you apply basic concepts of the lecture in python notebooks. - The seminar will have students read, critique, and present relevant deep learning research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The project will have students apply and design their own (small) deep learning project in the context of scientific reproduction.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the different deep learning techniques reviewed in the course, such as SGD, MLPs, CNNs, RNNs, GANs. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [LO4]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (eg: Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) problem domain which may need to re-implement, run, evaluate, investigate, extend existing research or code [LO8] Write clearly and concisely about your code, method, results, and analysis.	
Education Method	Lectures for basic theory based on the literature Assignments: we help you to become familiar with PyTorch; applying concepts from the lecture on small problems. Lab project: design and execute your own deep learning project in the context of scientific reproductions using https://reproducedpapers.org/ Seminar: paper reading, critiquing, and presenting.	
Literature and Study Materials	Books: freely available online: - http://www.deeplearningbook.org/ - https://d2l.ai/ Research papers will be made available through Brightspace. Assignments are based on PyTorch: https://pytorch.org/	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Project: in a small group of students you work on a deep learning paper reproducibility project. 3. Exam about the papers, assignments, and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4245	Seminar Computer Vision by Deep Learning	5
Responsible Instructor	Dr. J.C. van Gemert	
Instructor	S. Pintea	
Contact Hours / Week x/x/x/x	0/0/0/8	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Required for	MSc thesis in the Computer Vision lab	
Expected prior knowledge	Deep Learning (CS4240)	
Course Contents	The recent boom in computer vision and automatic image understanding represents an inflection point in human productivity, permeating wide aspects of the economy and society. Examples of visual tasks which are repetitive or require expert knowledge include medical diagnosis, industrial inspection, autonomous vehicles, etc. When machines can meaningfully assist or even completely take-over such tasks it will change the world as we know it. The breakthrough in the 2012 ImageNet automatic image recognition competition shows all previously existing methods decisively defeated by a deep neural network. Deep learning replaces feature engineering methods and is able to successfully learn image features from huge annotated datasets. This course is on automatically understanding visual content such as images and videos by deep learning. Topics include: Fundamentals in Vision, object detection, per-pixel labelings, video recognition, image similarity learning, efficiency, self-supervision, 3D computer vision, adverserial attacks, explainability, generative models. The course will have lectures, a seminar and a lab practical: - The lectures will be on established topics based on the current literature. - The seminar will have students read, critique, and present relevant computer vision research papers. You will have to read 2 papers per week, for 7 weeks (14 papers). - The lab will have students apply and design their own (small) computer vision project. The course build on top of the Deep Learning course (CS4240) and follows a similar setup.	
Study Goals	Upon successful completion of the course, students will be able to: [LO1]. Describe the deep learning techniques reviewed in the course for computer vision applications such as image classification, object detection, per-pixel labelings, video recognition, image similarity learning. [LO2]. Research literature concerning one of the above techniques, summarize it and report it to your peers [LO3]. Debate upon positive and negative aspects of techniques and research papers [L04]. Quickly identify the core contributions of a research paper [LO5]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (we focus on Pytorch) [LO6]. Determine which technique(s) is most appropriate for being used in a certain problem domain. [LO7]. Apply the appropriate technique to a (simple) Computer Vision problem. [LO8]. Write clearly and concisely about your code, method, results, and analyis.	
Education Method	Lectures Lab project: design and execute your own Computer Vision project. Seminar: paper reading, critiquing, and presenting.	
Assessment	1. Presentation: during the seminar a small group of students presents a paper. You will have to present once. 2. Students will have to submit relevant questions about papers/lectures 3. Lab assignment: in a small group of students you work on a deep learning project. 4. Exam about the papers and the theory. *** Disclaimer: Assessment this year may change depending on the COVID19 virus ***	

CS4250	Selected Topics in Molecular Biology	5
Responsible Instructor	Dr. T.E.P.M.F. Abeel	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1	
Course Language	English	
Course Contents	This course covers a broad range of essential topics in molecular biology necessary to start critically working with bioinformatics algorithms. The choice and implementation of algorithms critically depend on a proper understanding of the application domain. In this case the application domain is biology and in this course we review key biological concepts and topics that have impact on the implementation and selection of various computer algorithms and models. We use the following book: B. Alberts et al., 'Molecular Biology of the Cell', 6th edition Publication Date: November 18, 2014 ISBN-13: 978-0815344322 ISBN-10: 0815344325 A detailed overview of the actual reading material will be provided through the online learning platform Brightspace and will include material from Chapters 1,3,4,5,6,7 and 8. Extra material to aid in learning are available from the publisher's website. Covid-19 disclaimer: all information here is correct at the time of writing. This may change based on developments around COVID-19. Students will be informed as soon as possible through Brightspace.	
Study Goals	The goal of this course is to learn about the basic concepts in molecular biology required for bioinformaticians. At the end of this course: students will be able to reproduce, discuss and reflect on information about basic molecular biological processes, concepts and ideas. This includes: - cell structure, organization, information encoding and transfer - core concepts around DNA, chromosomes, genomes - DNA replication, repair and recombination - central dogma molecular, including replication, transcription, translation - regulation of the processes above	
Education Method	Self-study with weekly Q&A sessions.	
Books	B. Alberts et al., 'Molecular Biology of the Cell', 6th edition Publication Date: November 18, 2014 ISBN-13: 978-0815344322 ISBN-10: 0815344325	
Assessment	Oral exam and assignments by appointment disclaimer: information may change depending on the developments around the coronavirus.	

CS4255	Algorithms for sequence-based Bioinformatics	5
Responsible Instructor	Dr. T.E.P.M.F. Abeel	
Contact Hours / Week x/x/x/x	0/0/8/0	
Education Period	2	
Start Education	2	
Exam Period	2	
Course Language	English	
Course Contents	Covid-19 disclaimer: all information here is correct at the time of writing. This may change based on developments around COVID-19. Students will be informed as soon as possible through Brightspace. Bioinformatics analyses in genomics aim to compare large sets of genomes in order to understand and explain differences in traits of an organism. Contemporary methods are powered by fundamental algorithms and data structures, which are efficient and scale to large data sets. A thorough understanding of these algorithms and data structures is necessary for advanced users and developers in this area. In addition, understanding how comparative genomics is developing is important to shape your own research. In this course, we will cover genome analysis, variant analysis, and pangenomics. Core concepts, applications, and future trends will be discussed, with a focus on the algorithms and data structures underlying state-of-the-art methods.	
Study Goals	After having followed this course, the student has a good understanding of algorithms and data structures in genomics used for DNA sequence analysis. The student is able to implement algorithms in python, and can translate methods described in scientific literature into a working implementation.	
Education Method	The course is offered as a mix of lectures, exercises and a project	
Books	http://bioinformaticsalgorithms.com	
Assessment	60% graded exercises, 40% project. A resit opportunity is available for both components Students must receive a 'pass' mark for each component individually (>=5.8). Notes: - No scores and/or submissions are transferred to the next year or from previous years. - Rules and assessment format may change due to COVID19-regulations. This will be communicated as soon as possible. disclaimer: information may change depending on the developments around the coronavirus.	

CS4260	Machine Learning in Bioinformatics	5
Responsible Instructor	Dr. J.S. de Pinho Gonçalves	
Contact Hours / Week x/x/x/x	0/0/4/0 Lectures + 0/0/4/0 lab	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Recommended for fundamental knowledge in machine learning and multivariate data analysis: CS4220 Machine Learning 1 CS4070 Multivariate Data Analysis Recommended for basic concepts of molecular biology: CS4250 Selected Topics in Molecular Biology Required: CSE1200/TI1106M Calculus (or similar) CSE1205/TI1206M Linear Algebra (or similar) CSE1210/TI2216M Probability Theory and Statistics (or similar) The project requires basic programming skills.	
Course Contents	Learning from patterns in molecular biology data plays an important role in diagnosing disease, discovering new targets for therapy, and more generally in answering biological questions that lead to an improved understanding of biological systems with relevance to human health and industrial areas including biotechnology and agriculture. This course focuses on methodology for the analysis of high-dimensional data in molecular biology, naturally addressing challenges that commonly arise in the field such as learning from unlabelled data or from small numbers of samples. The methodology is introduced in the context of meaningful applications in molecular biology with examples using real data. Covered topics will be (a selection of) the following: - Discrete probability models and statistical modeling: statistical models for experiments with categorical outcomes, goodness of fit, maximum likelihood and Bayesian estimation. - Mixture models: finite mixture models of normals and the EM algorithm, common infinite mixture models such as beta-binomial and gamma-Poisson, ECDF and bootstrapping. - Clustering: comparing observations, iterative partitioning, density-based clustering, hierarchical clustering, clustering validation, choosing the number of clusters. - Statistical hypothesis testing: p-values, single and multiple hypothesis testing, family-wise error rate, (local) false discovery rate. - Testing using high-throughput count data: multifactorial designs, linear models, analysis of variance, generalized linear models, robustness and outlier detection, shrinkage estimation. - Linear dimensionality reduction (PCA): preprocessing data for multivariate analysis, projecting onto lower dimensions, matrix decomposition (SVD), biplot representations, projecting additional variables for interpretation). - Multivariate methods for heterogenous data: orderings, gradients and latent variables (ordination); multidimensional scaling (MDS), robust (non-metric) MDS, batch effect removal, correspondence analysis, finding gradients and trajectories (local non-linear methods such as tSNE and UMAP), canonical correlation. - Supervised learning: discrimination, performance measures, curse of dimensionality, generalizability and model complexity, regularization and penalization, cross-validation, supervised learning methods (SVM, decision trees, ...), method hacking. - Design of high-throughput experiments and their analyses: types of variability (error, noise, bias), confounding, dependencies, batch effects, statistical power, mean-variance relationships and data transformations, workflow design, data representation, efficient computation.	
Study Goals	After successfully completing this course, the student should be able to: - Recognise, characterise, and interpret different kinds of high-throughput molecular biology data and their statistical properties. - Recognise, categorise and compare common statistical techniques and machine learning algorithms and the data analysis problems that they address. - Recognise typical research questions that can arise when analysing such kinds of data, reason about and select appropriate methodology to address them. - Understand, reason about, and discuss the different steps of a data analysis workflow: from experiment design to the interpretation of results. - Design a research plan, execute it, and write a scientific paper about it.	
Education Method	The course is run in a flipped classroom setting. Lectures: students take turns explaining the material from the different chapters of the course book. All students are required to read and prepare the course material before every lecture, so that they can contribute to the discussion of the material in the classroom. Project/labs: students will work in small groups on a research project throughout the course. Each group will present and discuss the status of the project weekly or bi-weekly, and deliver a written report at the end. There will be one or two lectures (90 min. each) and one project discussion lab (45 to 90 min.) per week. The frequency and duration of the sessions will depend on the number of students following the course. The exact schedule will be determined after the first lecture.	
Books	The material of the course follows the book: "Modern Statistics for Modern Biology" Authors: Susan Holmes, Stanford University, California Wolfgang Huber, European Molecular Biology Laboratory Published: February 2019 ISBN: 9781108705295 The complete book can be browsed online at: http://web.stanford.edu/class/bios221/book/. This website also contains the R code and data accompanying the examples in the book.	

Assessment	- 25% Book chapter presentation - 15% Participation - 60% Project report/paper Students are graded individually. A passing grade for this course can only be obtained if the grade of every component except participation is at least 5.8. A resit opportunity is available for the project component. Grades of individual components are not carried over to future editions of the course. Disclaimer: information may change depending on developments around COVID-19.

CS4290	Seminar on Distributed Machine Learning Systems	5
Responsible Instructor	Dr. Y. Chen	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	none	
Course Language	English	
Course Contents	Machine learning systems are often conventionally designed for centralized processing in that they first collect data from distributed sources and then execute algorithms on a single server. Due to the limited scalability of processing large amount of data and the long latency delay, there is a strong demand for a paradigm shift to distributed or decentralized ML systems which execute ML algorithms on multiple and in some cases even geographically dispersed nodes. The aim of this seminar course is to let students learn how to design and build distributed ML systems via paper reading, presentation, discussion, and project prototyping. We provide a broad overview on the design of the state-of-the-art distributed ML systems, with a strong focus on the scalability, resource efficiency, data requirements, and robustness of the solutions. We will present an array of methodologies and techniques that can efficiently scale ML analysis to a large number of distributed nodes against all operation conditions, e.g., system failures and malicious attacks. The specific course topics are listed below. The course materials will be based on a mixture of classic and recently published papers. For each topic, the basic concepts and technology landscape will be first provided and then two state-of-the art of papers will be presented and discussed by students. We offer a testbed of a distributed (deep) ML system in which students are encouraged to apply different techniques to jointly improve its scalability and resilience. Course topics include Overview of distributed machine learning systems Performance and scalability of state-of-the-art systems Acceleration of ML workloads Slim distributed ML systems on small data Robust deep learning systems Federated machine learning systems	
Study Goals	Students are able to argue and reason about distributed ML from a systems perspective. Students understand the behavior and tradeoffs of distributed ML in terms of performance and scalability. Students can estimate the importance of data inputs via different techniques, i.e., core set and decomposition methods, for distributed ML systems. Students understand data poison attacks and design defense strategy for distributed ML systems. Students can analyze the state-of-the art federated machine learning systems and design the failure-resilient communication protocols. Students are able to design and implement methods and techniques for making distributed ML systems more efficient.	
Education Method	Lectures: 7 weeks X 2h Papers: one paper presentation, two paper reviews, and paper discussion. Practical: apply system and algorithmic optimization techniques learned in the lecture to improve the performance of distributed machine learning systems, e.g., image recognition on CIFAR 10. The testbed environment, learning algorithms, and dataset will be given. Deliverables include git commit of functioning code and a report summarizing the contribution	
Assessment	Paper presentation by group (10%): each group of 2 to 3 students needs to choose from a given set a papers to present (15 minutes) and lead the discussion (10 minutes). Paper reviews by individual (30%): each student needs to write three reviews of papers assigned from a given set and those two papers have to be different from the paper for the presented. Each review will account for 10% of the grade. Questionnaires by individual (0%): each student needs to hand in a list of questions at the beginning of the lectures that have paper presentation from other students. This is not graded. Individual project (60%): The project is collaborative among the entire class and competitive as a whole group. The objective is to continuously improve the performance of the given distributed ML system. The students need to hand in a final project report in style of a short scientific paper, stating their individual contribution to the overall system performance. disclaimer: information may change depending on the developments around the coronavirus.	

CS4329	Recent topics in bioinformatics	5
Responsible Instructor	Prof.dr.ir. M.J.T. Reinders	
Instructor	Dr. J.S. de Pinho Gonçalves	
Instructor	Dr. T.E.P.M.F. Abeel	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Bioinformatics is at the heart of many modern systems biology analyses, and encompasses the application of statistics and computer science to (large-scale) biomolecular datasets. In essence, bioinformatics is about smart ways of extracting knowledge from the enormous amounts of data that can be generated using modern measurement techniques. For instance, it plays an important role in finding the genetic origins of various diseases, such as cancer, diabetes or alzheimer. Covid-19 disclaimer: all information here is correct at the time of writing. This may change based on developments around COVID-19. Students will be informed as soon as possible through Brightspace.	
Study Goals	After successfully completing this course, the student is able to: explain several high-throughput data acquisition experiments, such as DNA/RNA sequencing, and discuss the benefits and limitations of these methods comprehend the statistical and computer science issues in analyzing high-throughput data discuss the basic systems biology approach, and the role of high-throughput measurements, gene selection and classification therein explain bioinformatics methods, algorithms and models to a non-expert audience implement or execute basic algorithms from descriptions provided in scientific literature read and comprehend a current scientific paper and reflect on the bioinformatics methods used in such paper	
Education Method	In this course we will study some key examples of bioinformatics analyses by reading a set of selected papers that present some significant biological conclusions.The course is run a flipped class-room students take turns guiding us through the selected papers. Teachers moderate the discussion and fill in any gaps in understanding.All students are required to read, and prepare the course material before every lecture to effectively participate. Each week there are one or two lectures, each of 90 minutes In each lecture one paper (the course material) will be discussed in detail. One or more students will present and explain the details of this paper. It is essential that you highlight the (bioinformatics) methodology of the paper. The schedule for this will be prepared in the first lecture. This presentation is graded. All students are expected to have read the paper and should have an active role in the discussion about the paper. Each student should be prepared to raise at least two critical remarks or questions during the discussion. The quantity and quality of participation of each student is graded. The student will perform a practical project in groups where they gain practical experience in state-of-the-art bioinformatics methods.	
Assessment	- Presentation (20%) - Participation during discussions (20%) - Reporting about unseen paper with oral presentation (60%) disclaimer: information may change depending on the developments around the coronavirus.	

EE4C06	Networking	5
Responsible Instructor	Prof.dr.ir. P.F.A. Van Mieghem	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	PART 1: Basics, concepts and computations of networks 1. Basics of networking & introduction to Network Science - what is a network? Representation of a graph, basics of graph theory, overview of the relatively new theory of complex networks, called Network Science. - important characterizers of a network (network metrics) - basic network/graph models - examples of real-world networks (airline transportation, the web and Internet, social networks, brain networks, etc.) and applications of network science 2. Concepts of networking - routing - Quality of Service (QoS) - traffic management and scheduling - network robustness (failure, cascading effects,...) - overlay networking and new aspects of networking such as interdependent networks PART 2: Applications and examples of networks (as listed below) will be taught (some of those by a guest lecturer). Ranging from year to year, a selection among the following will be covered: 1. Electrical networks (smart grids) 2. Networks on Chip (NoC) 3. Optical networks 4. Computer Networks (the Internet) 5. Mobile communication networks 6. Sensor networks 7. Biological networks 8. Social networks	
Study Goals	The course on Networking aims to provide a general and basic introduction to the art of networking, that tries to unravel the operation and behavior of networks, both man-made (infrastructures such as the Internet and power grids) as well as networks appearing in nature (such as the human brain, biological networks and social human interactions). The course on Networking will introduce concepts of the new Network Science, that basically studies the interplay between, on the one hand, the processes (also called functions or services) on the network and on the other hand, the underlying topology, that is mostly changing over time as an evolving organism, rather than as given or fixed object. Network Science combines many disciplines such as graph and network theory, probability theory, physical processes, control theory and algorithms. After this course, students are expected to represent/abstract real-world infrastructural network (e.g. a communication system) as a complex network, understand the basic methods to analyze properties of networks and dynamic processes on networks. Students will also understand why processes on networks and design of networks are so complex. Finally, students may appreciate the fascinatingly rich structure and behavior of networks and may realize that much in the theory of networks still lies open to be discovered.	
Education Method	Lectures, slides & homework	
Assessment	written examination disclaimer: information may change depending on the developments around the coronavirus.	

IN4010(-12)	Artificial Intelligence Techniques	6
Responsible Instructor	Dr. F.A. Oliehoek	
Instructor	J. He	
Instructor	Prof.dr. M.T.J. Spaan	
Instructor	Prof.dr. C.M. Jonker	
Contact Hours / Week x/x/x/x	3/3/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	1 2 3	
Course Language	English	
Expected prior knowledge	We expect students to have programming expertise at the Bachelor level of Computer Science, in particular knowledge is expected of algorithms (e.g. search algorithms), and probability theory (TW2215TI). Programming skills are required for the practical assignments which can be done in Java or Python, although some Java skill is expected.	
Course Contents	Artificial Intelligence techniques for building cognitive agents, and decision making and -support systems are presented. Techniques discussed include probabilistic reasoning, action selection and planning, Markov Decision Processes, reinforcement learning, and some other aspects of machine learning, as well as a variety of topics dealing with multiagent systems and strategic interactions.	
Study Goals	After successful completion of the course: - Students have a general overview of decision-theoretic artificial intelligence techniques - Students understand the working of the artificial intelligence techniques discussed - Students are able to apply the formal models covered in the course to new problems - Students are able to design, implement and evaluate algorithms for complex decision making problems.	
Education Method	Lectures, tutorials, lab work (practical assignments).	
Literature and Study Materials	Stuart J. Russel and Peter Norvig (2010). Artificial Intelligence: A Modern Approach. 3rd Edition. Prentice-Hall. ISBN-13: 978-0-13-604259-4 + additional handouts. See website http://aima.cs.berkeley.edu/ for additional information that goes with the book.	
Assessment	Written exam and practical assignments. The grade of the course is determined by the grade of the written exam (80%) and the mean grade of the practical assignments (20%). There are no resit possibilities for the practical assignments. There will be a resit possibility for the written exam in Q3.	
Remarks	The practical assignments will be done in groups of 4 to 5 students. The ethical standards of working are expected of all students. Work can be divided over students, but all students are responsible for the overall quality and originality of the work. Students that do not do their share of the work have to be reported by the group to the supervisor so that action can be taken.	

IN4049TU	Introduction to High Performance Computing	6
Responsible Instructor	Prof.dr.ir. H.X. Lin	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear algebra (matrix and vector operations), numerical analysis (solution of a system of linear equations;some experience with a programming language (e.g., C) is preferred but not required.	
Course Contents	This course is intended for students who are interested in computing-intensive research. In the course, a number of algorithms that are being used within a diversity of research areas is considered. The scaling behaviour of these algorithms in case of an increasing problem size and/or an increasing number of processors, is analysed. Attention is paid to those aspects of computer architectures that are important to understand the resulting performance, such as the memory hierarchy and the interconnection network. By analysing a number of case studies (applications) with respect to their computing-intensive character, possible bottlenecks will be determined. Based on performance analysis, it will be indicated how the effect of those bottlenecks can be reduced. The goal is to learn how to get a high performance with the available hard/architecture. The lab exercises will be done on a cluster of computers, the DAS-5 system at TU Delft. The emphasis will be on designing efficient parallel algorithms and on the necessary optimalisation of the performance. During the lab exercises, the following types of problems will be elaborated on: a parallel Poisson solver, a parallel finite element simulation and a parallel N-body simulation. More information, such as handouts and slides, can be found the Brightspace.	
Course Contents Continuation	High Performance Computing, parallel programming, parallel algorithm	
Study Goals	1. Knowledge about high performance computer systems including parallel and distributed architectures, and programming models; 2. Basic knowledge about the concepts of data decomposition and parallel algorithms; 3. Knowledge about various high performance (numerical) methods and their parallelization; 4. Capable to implement parallel programs (using MPI) on cluster of computers and GPU (using Cuda); 5. Obtain some experience on performance analysis of parallel programs.	
Education Method	Lectures, computer lab exercise using MPI. As an option, answers to the bi-weekly quizzes can be handed in, and a maximum of one bonus point to the exam grade can be obtained.	
Computer Use	Lab exercises (mandatory): implementing (small) parallel programs with C, MPI and Cuda.	
Literature and Study Materials	Will be made available throughout the course and can be downloaded from the Brightspace.	
Assessment	Written exam (50%) + Lab work (50%). disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	Via Osiris	

IN4089	Data Visualization	5
Responsible Instructor	T. Höllt	
Instructor	Prof.dr. E. Eisemann	
Contact Hours / Week x/x/x/x	0/2/0/0 & lab	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Required for	Master course MKE/ST/DS	
Expected prior knowledge	IN2905-A/IN4152/CSE2215 Computer Graphics (recommended, not required). The practicals will be implemented in HTML/Javascript/D3 (InfoVis) and C++ (VolVis). We consider programming skills as a requirement but not necessarily in the mentioned languages and no advanced skills are needed. Relevant topics will be introduced and experience in other programming languages should make adaption feasible.	
Course Contents	Data visualization is the visual representation of data by computer generated images. The data sets can be results of numerical simulations or measurements (scientific visualization), or other data collections such as databases (information visualization). The goal is to improve insight, understanding and/or communication of data. Data visualizations use a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithmics, image processing, machine learning, numerical analysis, optimization, The course has two main parts; information and scientific visualization that will involve knowledge of diverse disciplines. As a computer science course, affinity to algorithmic thinking and programing skills will be needed. Topics covered: models of the visualization process; colour models and use of colour; information visualization; representation and processing of data; volume visualization; interactive visual data analysis; visualization of vector fields and flows. Guest lectures might be given on selected topics.	
Study Goals	The goal of the course is to get knowledge on the fundamentals that are part of data visualization. The main principles and techniques that are the basis of generating effective visual representations of data. Techniques and cases of data visualization are discussed. There are several applications for the techniques, such as medical, engineering, finances, economics, game analytics, and more. By the end of the course, you should be able to LO1: Discuss a large range of visualization techniques. LO2: Discuss a perception principle of visualization. LO3: Explain mathematical principles and algorithms of visualization techniques. LO4: Design suitable visualization systems for a given practical data analysis problem. LO5: Implement visualization systems for a given practical data analysis problem.	
Education Method	Lectures, practical assignments, self-study, and projects.	
Literature and Study Materials	Course slides, instructions for projects, and selected literature. Chapters from: Visualization Analysis and Design Author: Tamara Munzner CRC Press Visual Computing for Medicine 2nd Edition Theory, Algorithms, and Applications Authors: Bernhard Preim Charl Botha Morgan Kaufmann All available in electronic form via Brightspace or at TUDelft library.	
Assessment	The final grade is a weighted average based on two visualization projects, and a written exam that might contain multiple choice questions. The projects will be developed in groups of 1-3 and are evaluated based on the reasoning/justification of the techniques used based on the material given at the course, effectiveness of the results, technical contribution or implementation, quality of the documentation and presentation. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Judgement	The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam. The two projects will be developed in couples and will represent 70% of the mark together. All projects, which are handed in late will be evaluated with a zero and impact the part of the mark that corresponds to the project. Additionally, a written exam will be held, which will represent 30% of the mark. The exam might contain multiple-choice questions. The project is evaluated based on the developed result, its documentation and presentation. Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The exam will have a resit. No resit will be provided for the projects unless the mark on the exam (NOT the resit exam) and the other project are above 7.5 The project resit is not automatic and must be initiated by you within two weeks of the grades being published. It will be evaluated at individual bases, despite the project being done in groups. Resit of a project will mean starting a new project.	

IN4150	Distributed Algorithms	6
Responsible Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Knowledge of Computer Networks (CSE1405) and Operating Systems (CSE2430) is useful as background for understanding this course.	
Course Contents	Introduction to distributed algorithms; notions of time and ordering of events; distributed algorithms for message ordering, detecting global states, termination detection, deadlock detection, mutual exclusion, election, minimum-weight spanning trees, fault tolerance, consensus, and agreement; blockchain technology and its relation with consensus.	
Study Goals	Understand the main fundamental problems in distributed systems Understand the most important distributed algorithms that solve these problems Be able to reason about the execution of distributed algorithms Be able to program distributed algorithms Be able to select and summarize relevant literature on distributed algorithms	
Education Method	Lectures and lab work executed in groups of two students	
Literature and Study Materials	Lecture notes and lecture slides (available on Brightspace)	
Assessment	One paper summary and an in-person, on-campus written exam (closed book), which together are an integrated whole. The grade of this exam, which includes the grade of the paper summary (with a weight of 25%), is the grade for the course. In case in-person, on-campus written exams are not possible because of COVID-19, the written exam will be replaced by individual online oral exams.	
Permitted Materials during Tests	None except the list of algorithms	
Remarks	Lab work is 40 hrs.	

IN4252	Web Science & Engineering	5
Responsible Instructor	J. Yang	
Responsible Instructor	Dr. C. Lofi	
Instructor	Prof.dr.ir. G.J.P.M. Houben	
Contact Hours / Week x/x/x/x	2/2/0/0	
Education Period	1 2	
Start Education	1	
Exam Period	none	
Course Language	English	
Expected prior knowledge	The expected entry level is (equivalent to) standard bachelor-level computer science.	
Course Contents	The main subject of the course is the Web, and in particular Web Data. The course considers developments in the Web and the (big) data management challenges associated to it. In particular, the course considers the relationship between people and technology that come with the Web and Web-based information systems. The course considers the Web both from an engineering perspective as well as from an analytical perspective. The course explains the concept of Web-based Information System and thus concentrates on a large class of modern information systems that use the web and web data in one way or another. The course gives an insight into the research area of Web Engineering, where methods and techniques for the design and development of web-based information systems are investigated. The course outlines the developments related to Web Data, and its management, processing and retrieval. The course gives an overview of the research and practice concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner. It also considers the concepts behind Linked Open Data and the data processing pipelines to create and analyse Linked Open Data. With the social-technical nature of the Web and its systems, the course pays attention to the interplay between people and systems. The course gives an overview of the research area of User Modeling, with its main approaches and techniques to represent and capture properties of users that provide a basis for user-adaptation and personalisation in web-based information systems. In relation to user modeling, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. The course also considers recent developments in the research area of Human Computation concerning the role of humans in the processing of (human-related) web data, for example using crowdsourcing to create or annotate web content. As the web and its data are mirroring the world and the people in it, the course also takes a look at Web Science, as a branch of data science that considers the largest human-made artefact ever, i.e. the Web, and how that analytical research is addressing a whole new range of challenges. These challenges include studying how data analytics can be done by means of Web data, as well as studying how new systems can be created and engineered to make use of the Web and its properties.	
Study Goals	The student learns the important principles and concepts of Web-based Information Systems, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used for Data Management in the area of web-based information systems, in particular concerning the Semantic Web and Linked Open Data. The student has knowledge of the main principles and techniques for User Modelling and adaptation, and of the role of Social Web data and Human Computation for user modelling. The student learns the major challenges and principles from the research in the field of Web Science, and the role of web data for Web Science. The student is able to write a paper contributing to Web Science based on a problem in the field of web-based information systems.	
Education Method	The education includes: - Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material; - Small assignments and hands-on exercises, to apply the understanding of relevant material; - One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research. Lectures will be not each week in the class period (1+2): in between lectures there is time reserved for studying before and after lectures, for small assignments and exercises, and for writing the large assignment paper. The writing of the large assignment paper happens throughout the class period (1+2) to enable frequent feedback.	
Literature and Study Materials	Will be provided in class, depending on the topics chosen for the assignments and final paper.	
Assessment	Assessment happens on the basis of the small assignments (accompanying the lectures), for 20% of the grade, and the large assignment (writing the web science paper), for 80% of the grade. All assignments must be completed by the indicated deadlines. Students must get a pass for each assignment. The final paper needs to be graded with a 6.0 or higher. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Students are asked to register/enrol on Brightspace. Students are also asked to be present and active in the first lecture session, to facilitate the proper planning of the course.	
Remarks	The expected workload is 5ects and that is principally distributed uniformly over the two quarters. The course is completed with a final paper; students have one chance per year to re-submit a paper if it is judged insufficient.	

IN4306	Literature Survey	10
Responsible Instructor	Dr.ir. W.P. Brinkman	
Responsible Instructor	L. Miranda da Cruz	
Contact Hours / Week x/x/x/x	Not applicable	
Education Period	None (Self Study)	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	The Literature Survey is an individual assignment carried out under the supervision of a Computer Science staff member, i.e. an assistant, associate or full professor. For this assignment the student reads a broad range of papers in the chosen specialisation field and writes a report in which the ideas found in the papers are discussed and compared. It is not allowed to merge this assignment with the thesis project.	
Study Goals	The student is able to read contemporary scientific literature in the chosen field of specialisation. The student is able to distill the main ideas of a paper and to write these down in his or her own words. The student is able to place the ideas of different papers in perspective by comparing these. The student is aware of the most important academic journals and conferences of the research field of the chosen specialization. The student understands the role of communication and writing inherent in academic research (e.g. peer review process). The student understands experimental principles (hypothesis, validation, evaluation, theoretical versus empirical results, .).	
Education Method	Individual assignment and individual guidance by a scientific staff member.	
Assessment	Writing a scientific report, individually and under supervision of a staff member. This staff member will also mark the report. disclaimer: information may change depending on the developments around the coronavirus.	
Enrolment / Application	The Literature Survey may be part of an individual exam programme of a student, which has to be approved by the Board of Examiners (BoE). To apply for a literature study the student should contact a staff member of the research group of her/his chosen specialisation after having received approval of his or her individual exam programme. The staff member and the student make arrangements regarding content and scope of the survey. The abovementioned staff member will supervise the student during this course.	
Co-Instructor	Dr.ir. A.R. Bidarra	
Co-Instructor	Dr. T.E.P.M.F. Abeel	
Co-Instructor	Dr. K.A. Hildebrandt	
Co-Instructor	Dr. J.C. van Gemert	
Co-Instructor	Dr. C. Lofi	

IN4307	Medical Visualization	5
Responsible Instructor	T. Höllt	
Contact Hours / Week x/x/x/x	2/0/0/0 lectures & 0/4/0/0 lab.	
Education Period	1 2	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic knowledge of linear algebra, calculus and programming is needed. This course (IN4307) has been designed to complement the courses Advanced Image Processing (ET4283) and Medical Imaging (AP2231TUD). However, these two courses are NOT pre-requisites.	
Course Contents	Theory and practice (Notice project extends to Q2) of medical visualization. This includes the following aspects: data acquisition basics, clinical practice; image processing, e.g., filtering, segmentation and measuremen; medical volume visualization; illustrative visualization; advanced visualization for complex modalities; interaction techniques for medical data; advanced applications.	
Study Goals	By the end of the course, you should be able to LO1: Explain medical visualization algorithms and their applicability to medical problems. LO2: Discuss the advantages and disadvantage of medical visualization algorithms. LO3: Build a medical visualization system for a given problem: a. Discuss a suitable visualization for a given medical problem. b. Implement the most suitable solution. c. Judge the performance of the implemented solution.	
Education Method	The course will be based on a combination of lectures and practical assignments. A final project will be developed in Q2	
Literature and Study Materials	Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications Bernhard Preim and Charl P. Botha (not mandatory)	
Assessment	The evaluation will be based on - a written (or oral if the number of students allows) exam (40%) - assignments during the semester (10%) - a final project (50%) The final project will be done during the 2nd quarter. Regarding minimal grades for partial examinations: A passing final grade for a course can only be earned when for all component examinations and practicals of that course at least a 5,0 is earned, and the average grade for all components is at least a 5,8 (Article 17 RRBE (subsection 6)) The assignments will consist of small programming exercises and open questions, as preparation for the final project. The practical sessions will provide time for working on the assignments with assistance. The deliverables will be program code and/or answers to questions. The final project will be the design and implementation of a visualization system for a given medical problem. The final project will be carried out in teams. The deliverables for the final project will be a report (paper), the results (e.g., code) and a short video presenting the project (i.e. screencast). The written exam will be arranged at the end of the first quarter. You are allowed to have the slides and material of the course during the exam. No computer or laptop is allowed. The exam has a resit. The project will have a resit if the exam (NOT the resit exam) has been passed with a mark of 7.5 or higher and it will be on an individual basis. The project resit is not automatic and must be initiated by you within two weeks of the grades being published. Resit of a project will mean starting a new project. disclaimer: information may change depending on the developments around the coronavirus.	
Permitted Materials during Tests	Notes and written material. No computers.	
Special Information	It is necessary that you register/enroll on Brightspace for this course. In the first lecture, details on the evaluation and practical information on the course will be given.	
Co-Instructor	Prof.dr. E. Eisemann	

IN4315	Software Architecture	5
Responsible Instructor	Prof.dr. A. van Deursen	
Responsible Instructor	Prof.dr.ir. D. Spinellis	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Software engineering	
Course Contents	The software architecture course offers students a chance to learn and experience the concepts of designing, modeling, analyzing and evaluating software design and software architectures. Furthermore, the course provides students with a discussion forum in which recent articles in the area of software architecture are presented and discussed. The course also features a number of guest lectures to show the state-of-the-art of software architecture in industry. Topics covered by this course are: fundamentals of software architectures, modeling and designing software architectures, architectural patterns and styles, architecture viewpoints and perspectives, the role of the software architect, analyzing and evaluating software architectures, component and plug-in frameworks, software product lines, service oriented architectures, code quality, technical debt, refactoring. The course includes extensive labwork in groups of four, in which the actual architectures of existing open source systems are analyzed in considerable detail. These systems are taken from github, and student teams are challenged to actually contribute to the systems under analysis in the course.	
Study Goals	Bring students into the position that they can (1) explain the key architectural concepts and methods for modeling software architectures; (2) apply viewpoints and perspectives to model software architectures; (3) discuss the benefits of architecting and the role of the software architect; (4) evaluate and validate software architectures; (5) explain and discuss the concepts of component-based and plugin architectures, service-oriented architectures, and software product lines; (6) explain and recognize technical debt and have an understanding of possible refactorings.	
Education Method	Interactive lectures, lab assignment, paper presentation and discussion.	
Literature and Study Materials	The course uses the books; Cesare Pautasso. Software Architecture: Visual Lecture Notes. Leanpub, 2020; and Coplien and Bjørnvig, "Lean Architecture", Wiley, 2010. Additional reading material will be announced in the lectures.	
Assessment	No written exams. Grades will be based on lab assignment including essay writing, coding, (video) presentation, peer reviewing, participation. Disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	Course web site: https://se.ewi.tudelft.nl/delftswa/index.html	
Co-Instructor	M. Finavaro Aniche	

IN4325	Information Retrieval	5
Responsible Instructor	Dr. C. Hauff	
Responsible Instructor	J. Yang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3	
Course Language	English	
Expected prior knowledge	Knowledge of basic algebra. Proficiency in at least one programming language. Knowledge of Web information systems and software engineering can be helpful.	
Course Contents	Retrieving relevant information is one of the central activities in modern knowledge-driven societies. As the amount and variety of data increase at an unprecedented rate, access to relevant, possibly unstructured information is becoming more and more challenging. The World Wide Web is now the primary source of information for leisure and work activities. The real value of the Web can only be unlocked if the huge amount of available data can be found, analysed, and exploited so that each user can quickly find information that is both relevant and comprehensive for their needs. Information Retrieval (IR) is the discipline that deals with the representation, storage, organisation of, and access to information items, and it is concerned with providing efficient access to large amounts of unstructured contents, such as text, images, videos etc. The field is closely related to Natural Language Processing (NLP) that offers state-of-the-art methods for processing and analysing textual data. The objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the fields of Information Retrieval and Natural Language Processing. The course aims at providing students basic information retrieval concepts and more advanced techniques for efficient data processing, storage, and querying. Students are also provided with a rich and comprehensive catalogue of information search tools that can be exploited in the design and implementation of Web and Enterprise search engines. Covered topics include: = Information Retrieval Models; = Indexing Techniques; = Web Search; = Information Seeking Paradigms; = Evaluation of information retrieval systems; = Components of Natural Language Processing; = Word Embedding Techniques; = Neural Language Models; = Deep Contextual Embeddings; = Neural Information Retrieval	
Study Goals	At the completion of this course, students will be able to: = Describe the different information retrieval models, and compare their strenghts and weaknesses. [Learning Objective 1] = Describe and implement different indexing techniques. [Learning Objective 2] = Describe and analyze querying techniques with respect to their most suited application domains. [Learning Objective 3] = Analyse the effectiveness of an information retrieval system through proper use of evaluation metrics. [Learning Objective 4] = Design and implement (Web) Information Retrieval systems, possibly using advanced social and semantic search functionalities. Support and defend the relevance and correctness the choices with regards to the adopted information retrieval model, indexing technique, and querying technique. [Learning Objective 5] = Describe typical NLP tasks and components of an NLP system. [Learning Objective 6] = Describe neural language models, and compare their strengths and weaknesses to N-gram models. [Learning Objective 7] = Compare word embedding and contextual embedding techniques. [Learning Objective 8] = Design, implement, and evaluate NLP systems for IR and other NLP tasks. Justify the choices of NLP techniques and analyze the pros and cons using suitable metrics and qualitative analysis. [Learning Objective 9]	
Education Method	Lectures; course long group project (research and development) as well as an individual literature survey and small weekly assignments. Expected workload is is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 80 hours for the group project and 15 hours for the literature survey.	
Literature and Study Materials	Scientific papers, course slides, course books - all resources are available on Brightspace.	
Books	Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA. Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press. Dan Jurafsky and James H. Martin. 2014. Speech and language processing. Pearson. Yoav Goldberg. 2015. A Primer on Neural Network Models for Natural Language Processing. Bing Liu. 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".	
Assessment	Weekly individual assignment, weighting 10% of the final grade. Two group projects for IR and NLP, each weighting 45% of the final grade. The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities. disclaimer: information may change depending on the developments around the coronavirus.	

IN4331	Web-scale Data Management	5
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Dr. A. Katsifodimos	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Expected prior knowledge	Bachelor level courses in database management systems and operating systems. A prior course in distributed systems or middleware would be helpful but is not required. Programming skills are important for the final assignment.	
Course Contents	This course addresses the challenges of Data Management at Web-scale. Especially, it covers the need for large-scale distributed data storage systems. The lecture therefore introduces step-by-step increasingly complex distributed storage systems, leading up to modern implementations of different NoSQL data storage systems. The challenges arising from such systems are presented and discussed, especially focusing on the CAP theorem and the resulting trade-offs with respect to data models, transactional power, query expressivity, and replication consistency. These discussions lead to different variants of NoSQL database systems, like Key-Value Stores, Document Stores, Wide-Columnar stores, and Graph Databases. The advantages, disadvantages, and general properties of these systems are discusses in more detail. There is special focus on distributed transactions and consistency guarantees of different data management systems and methods.	
Study Goals	At the end of this course the student can - assess the nature of a given storage problem, and can select a suitable technology for solving it - understands the different data models encountered in Web Data Management, and their impact on modelling and querying - understands the issues arising from distributing and replicating data, especially with respect to the CAP theorem -understands the trade-offs which can be chosen within the design space of the CAP theorem -categorize and explain modern NoSQL databases within the framework of the previously mentioned trade-offs	
Education Method	Lectures and assignments	
Literature and Study Materials	Course slides and Lecture Videos	
Books	Literature mentioned in the lecture, mostly research papers.	
Assessment	Depending on the year, there are group assignments with final presentation, written report, and oral interviews. Final grade will be the weighted average of the three partial grades (the weights will be announced during the lecture depending on the available topics and assignments in the current quarter). The interview covers questions from the topics of the lecture. The interview can be resit, the assignment & presentation cannot. disclaimer: information may change depending on the developments around the coronavirus.	

IN4344	Advanced Algorithms	5
Responsible Instructor	Prof.dr. M.M. de Weerdt	
Instructor	Dr. E. Demirovi	
Instructor	Prof.dr.ir. K.I. Aardal	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Basic courses in Algorithmics and Complexity Theory	
Course Contents	The course is on solving (abstract models of) complex real-world problems, with a focus on solving intractable problems exactly. The course consists of two main parts: * modeling and solving using linear programming * exact algorithms using search trees, dynamic programming, and/or decision diagrams	
Study Goals	By the end of this course, students will have knowledge of and experience with the following advanced algorithmic techniques: (Part 1:) - linear programming (LP) and LP modelling - duality and simplex algorithm - integer LP and integer modelling - LP relaxation and branch and bound (Part 2:) - complete and bounded search trees - tree decomposition - dynamic programming - preprocessing - decision diagrams Furthermore, they will be able to * recognize situations where they can use these techniques and adapt them to different applications. construct new algorithms that are similar to these techniques. construct proofs that are similar to a selected set of proofs from the material. analyze the run time of algorithms. implement an algorithm that is given in pseudo-code. experimentally evaluate the quality and the runtime of an algorithm on a set of instances.	
Education Method	Interactive lectures, optional homework exercises, programming assignments The expected workload is 30% studying the written material and the recorded lectures, and participating in the interactive sessions 30% making the homework exercises and preparation for the exams 40% working on the programming assignments	
Literature and Study Materials	Part 1 of the course will be mainly based on chapters 1-8, some of 9-11 (most of 9-11 is assumed to be known), and 12-14 of the syllabus "Optimization" by Karen Aardal, Leo van Iersel and Remie Janssen, which can be ordered via https://www.webedu.nl/bestellen/tudelft/ Part 2 will use chapter 10 of the following textbook: J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education, 2006. ISBN 0-321-37291-3 Supplemental study material will be provided via Brightspace.	
Assessment	The final mark depends on the marks obtained for (a) programming assignments, PA (weight 40%) (b) the exam, EX (2 parts, weight 60%) Each programming assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is the average of the mark obtained for the assignments. Programming exercises can be completed by 2 students working together. The exam consists of two parts. Each part will be examined after the lectures about that part have been delivered and will be graded on a scale from 1 to 10. The final mark for the exam (EX) is the average of the marks for the parts. Each exam part contains one or more challenging assignments; during each part of the course two representative homework assignments will be made available to prepare for this. There is a resit for the exam where any of the two parts can be redone. The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. There is no repair option for the programming assignment. The final mark for the course is determined as follows: - if the PA and the EX mark are above 5, the final mark is the weighted average of these three marks: 60% EX, 40% PA - if at least one of PA, EX is less than or equal to 5, the final mark is the minimum of the results obtained for PA or EX. Partial results are valid only in the current academic year. disclaimer: information may change depending on the developments around the coronavirus.	
Special Information	For content questions / of general interest, please use StackOverflow. For personal questions, please use the course email address: aa-cs-ewi@tudelft.nl	
Tags	Algorithmics Artificial intelligence Mathematics	

IN4391	Distributed Systems	5
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Instructor	Prof.dr.ir. D.H.J. Epema	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Starting with the mid-1990s, computing is undergoing a revolution, in which collections of independent computers appear to users as a single, albeit distributed, computing system. Motivated by the increase in the computation capacity of consumer computers, by the commoditization of server-grade machines, and by the advent of the Internet, the distributed computing paradigm has permeated all fields using computers. Current distributed computing applications range from the consumer social networks, peer-to-peer file-sharing, and massively multiplayer online games; to scientific computing using Big Data and distributed sensors; and to engineering fields and industrial control systems. This course focuses on the systems aspects of distributed computing. Specific, contemporary distributed systems are used as illustrative examples to discuss system design and non-functional requirements.	
Study Goals	1. Explain the objectives and functions of distributed computing systems. 2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems. 3. Describe the architecture and operation of distributed computing systems. 4. Explain how distributed computing systems can process user workloads. 5. Explain how distributed computing systems can detect and correct faults and errors. 6. Implement complex operations of modern distributed computing systems in realistic scenarios. 7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)	
Education Method	Lectures: This class uses a partially flipped classroom setting. Students are required to prepare parts of the material through self-study to be able to follow the in-class discussion. This involves, e.g., reading scientific papers. Practical: Designing, implementing, and evaluating a complete distributed system in groups, based on existing research work. Multiple topics are given to choose from. Deliverables include the code and a report of max. 10 pages.	
Literature and Study Materials	Textbook: Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Principles and Paradigms (2nd Edition), Prentice Hall, 2006. The textbook introduces the student to the traditional theory of distributed systems. Additional material: Several relevant research articles introduce the student to the classic literature as well as the latest advances on the topic.	
Assessment	Written exam (closed-book, open questions) during the exam period. A resit for the exam is offered in the following exam period. Practical project assessed based on the code, a presentation, and the report. This course uses gamification. Points can be collected through the practical project (max 4000 points) and the final exam (max 6000 points). The final grade is determined proportional to the 10000 points total. disclaimer: information may change depending on the developments around the coronavirus.	

IN5000	Final Project	45
Responsible Instructor	Dr.ir. A.R. Bidarra	
Responsible Instructor	Dr.ir. W.P. Brinkman	
Responsible Instructor	Dr. K.A. Hildebrandt	
Responsible Instructor	Dr. J.C. van Gemert	
Responsible Instructor	M. Finavaro Aniche	
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Contact Hours / Week x/x/x/x	x/x/x/x	
Education Period	1 2 3 4 Summer Holidays	
Exam Period	none	
Course Language	English	
Course Contents	IN5000 Final Project is the final part of the Master's degree programme. During this project, you will be required to demonstrate your ability to solve a research or engineering problem. The project must be carried out using the techniques of project management. You will begin by making a project plan in cooperation with your Masters thesis advisor. Several aspects of the project are defined within the plan, including the assignment, the frequency of interaction with the advisors, the milestones of the project and the resources and facilities offered by the faculty. You will be required to adhere to your plan throughout the project. It is obviously possible to adjust your plan under certain circumstances and after discussion with your daily supervisor. At the end of the project, you will submit your Masters thesis, which must be written in English, and make an oral presentation of your work to the Thesis Committee. The Thesis Committee will announce the final mark, which is based on the project performance, the thesis, the presentation and the subsequent discussion. More information about the graduation process: https://www.tudelft.nl/en/student/faculties/eemcs-student-portal/education/graduation-policy-msc/	
Study Goals	1. The student is able to design a research project: - The student is able to use, explain and justify adequate research and design methodologies; - The student is able to apply theory to the performed project; - The student is able to use techniques for interpretation and verification and bases his/her conclusions on results; - The student is able to do reliable work with scientific significance. 2. The student is able to execute a research project: - The student has a critical attitude towards his/her own results, literature and specialists; - The student makes an original contribution to the project; - The student takes initiative (together with the supervisor) to give his/her own input within the research project; - The student interacts sufficiently with peers and superiors; - The student is able to make and execute a project plan. 3. The student is able to write a research report: - The student is able to write a research report that shows sufficient coherence of content; - The students is able to structure the research report and sufficiently present the content (text and figures); - The student expresses argumentation using correct spelling and grammar; 4. The student is able to present and defend the research project: - The student is able to present the content using sufficient detail to support conclusions; - The student is able to logical structure the presentation and use visual aids; - The student is able to adequately formulate and express himself/herself as well as sufficiently address the audience; - The student is able to argument and answer the questions asked by the committee.	
Education Method	Project	
Prerequisites	Before starting the project, students must have completed at least 60 EC of the Master's degree programme and be in possession of a Thesis Entrance Permit (TEP). To be able to get a TEP, the individual exam program (IEP) should have been approved by the Board of Examiners. Note: In some cases, the thesis supervisor may impose additional conditions for starting this project.	
Assessment	The thesis committee assesses the thesis and the defense on the following criteria: - quality of work: novelty, volume, grasp, methodology, publishable; - personal performance: autonomy, planning, creativity, attitude; - quality of thesis report: clarity, organisation, argumentation; - oral presentation and defense: clarity, focus, relevance, discussion. More information on thesis grading: https://www.tudelft.nl/en/student/faculties/eemcs-student-portal/education/graduation-policy-msc/assessment/ The voting members of the thesis committee determine the final grade. The grade should reflect a weighted average of the four scores above, but need not to be an exact arithmetical mean. The final mark starts from 5 up to and 10. Marks ending in .5 may also be used. If the student shows excellence (is nominated for a 10) the chair of the thesis committee should consult the chair of the Board of Examiners, at least five working days in advance of the defense. The chair may advice to add an extra member to the thesis committee. The motivation for the grade at each of the four criteria as listed above is summarized on a form and signed by the chairman of the thesis committee. The candidate is given a short account of the assessment, either in private or in front of the audience. disclaimer: information may change depending on the developments around the coronavirus.	

NB4130TU	Biologic	3
Responsible Instructor	Dr. H.J.E. Beaumont	
Contact Hours / Week x/x/x/x	This course is not taught in 2021-2022	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	>>This course is not taught in 2021-2022<< *Students from all MSc programs can join* *Fully online course with interactive sessions, lectures and biodesign group-project* *Open for enrolment on Brightspace now* *No previous background in biology beyond high school-level required* Life has evolved solutions to an extraordinary range of problems and can itself be engineered to perform new functions. This has already inspired and enabled a broad range of technological innovationsfrom wind power and robotics to waste water treatment, architecture, computer science and beyond (see for examples: https://asknature.org/)but we have only begun to scratch the surface of what biology has to offer. To be able to effectively discover biological solutions and translate them to technology, engineers and designers need an understanding of the life sciences and insight into the biomimicry approach. BioLogic breaks down the traditional academic boundaries to kick start your ability to draw inspiration from biology. You will study key concepts in biology in relation to examples of bio-inspired engineering and design. You will also learn bio-inspired engineering and design methodology. In parallel, your biomimicry skills will be developed in a group project in which you will create a novel bio-inspired innovation in a field of your interest. The course features online live interaction sessions, live and recorded lectures, reading of a university-level biology textbook, reading quizzes, a group project involving writing of a (brief) proposal on your bio-inspired innovation and live online group-work coaching sessions.	
Study Goals	-Understand a step-wise biomimicry approach for design and engineering -Access and interpret educational and scientific sources of biological information -----Understand the universal hierarchical architecture of living systems ----Understand the key building blocks of life at all levels of biological organisation (incl. the level of genes, proteins, proteinnanomachines, cells, tissues, organs, viruses, bacteria, fungi, animals and plants, unicellular life, multicellular life, populations, ecosystems and biosphere) ----Understand the key processes of life at different levels of biological organization (incl. gene regulation, protein synthesis, energy metabolism, sensing at the cell level, rotary motors at the nanoscale, cellular decision making networks, photosynthesis, embryonic development, cell-cell communication, physiology, reproduction, self organisation, self repair, evolution, ecosystem dynamics, geochemical ecological cycles, ecosystem resilience). -Generate initial solutions to design- and engineering problems following a step-wise biomimicry approach: ---Analyse technical challenges and identify and evaluate potential biological solutions ---Translate biological solutions to design and engineering concepts ---Develop a novel bio-inspired or bioengineering innovation in a TU Delft discipline -Understand state of the art examples of bio-inspired design/engineering cases (incl. artificial intelligence, photosynthesis based energy conversion systems, urban design, solar power, wind power, airplane engineering, waste water treatment, robotics, smart materials, medical devices, architecture, computer science, human computer interfaces). -Understand basics genetic bioengineering and its applications (incl. microorganism-based wastewater treatment, synthetic life, genetic engineering, CRISPR-Cas DNA engineering, biofuel production, biological fine chemical production, directed evolution, DNA origami, artificial intelligence, protein nanomachine engineering, self organizing multicellular systems).	
Education Method	Lectures, textbook reading, reading quizes, group project, short project document, short presentation	
Books	"Biological Science" 6th edition by Freeman et. al (Global Edition, Pearson) Ebook version is available: https://www.vitalsource.com/en-uk/products/biological-science-global-edition-scott-freeman-v9781292165080	
Assessment	Group project grade (optionally adjusted with Synergy factor when team-member evaluations vary substantiialy) 50% and written exam 50%. Passing grade for the written exam is required to pass the course.	
Remarks	This course is not taught in 2021-2022	
Elective	Yes	

Year	**2021/2022**
Organization	**Electrical Engineering, Mathematics and Computer Science**
Education	**Master Computer Science**

Cyber Security 2021	
In association with the Faculty of	TBM
Introduction 1	The 4TU.CybSec master specialisation program offers a large variety of courses enabling students to select a highly individual course program. See also the website https://www.4tu.nl/cybsec/en/course-program/ Student in the special programme in Cyber Security may choose between the Data Science & Technology, the Software Technology track and the Artificial Intelligence Track. Students do at least 120 ECT in total including 80 ECT Cyber Security. I Common Core Courses Special Programme Cyber Security 1. IN4191 Security and Cryptography, 5EC 2 CS4035 Cyber data analytics, 5EC 3. SPM5442 Cyber risk management, 5EC 4. CS4150 Systems Security, 5EC 5. ET4397IN Network Security, 5EC Students have to complete two additional common courses from their respective tracks II Technical Electives: choose at least 3 courses 1. IN4253ET "Hacking Lab"-Applied Security Analysis, 5EC 2. UT201500040 Introduction to Biometrics, 5EC 3. UT201500042 Privacy Enhancing technologies, 5EC 4. AP3421 Fundamentals of quantum information, 4EC 5. CS4090 Quantum communication and cryptography, 5EC 6. CS4160 Blockchain Engineering, 5EC 7. UT192110940 Secure data management, 5EC 8. CS4110 Software Testing and Reverse Engineering, 5EC 9. UT201500039 Security verification, 5EC 10. CS4106 Dynamic and Static Programme Analysis for Software Security, 5EC 11. CS4265 Computer and Network Security: Advanced Topics, 5EC 12. UT202000026 Secure Cloud Computing, 5EC III Socio-Technical Electives: choose at least 3 courses 1. UT191612680 Computer Ethics, 5EC 2. UT201100022 Cyber crime science 5EC 3. UT201500038 E-Law, 5EC 4. WM0824TU Economics of Security, 5EC 5. UT201500041 Cyber security management, 5EC 6. CS4185 Capstone Cyber Security 5EC - Capstone CybSec Social skills - Capstone CybSec Entrepreneurial skills - Capstone CybSec Business skills 7. TMP4110A Governance of Cyber Security, 5EC IV Required Courses for CS Graduation 1. CS4120 Seminar Cyber Security, 5EC 2. IN5000 Master Thesis Project in Cyber Security, 45EC The thesis (45 credits) is performed under supervision of the Cyber Security research group. V Free Electives (Prerequisites and other Courses) The remaining credits to make up the programme are chosen in consultation with the master coordinator.
Administration by the Faculty of	EWI

AP3421	Fundamentals of Quantum Information	4
Responsible Instructor	Dr. L. di DiCarlo	
Instructor	Dr. D. Elkouss Coronas	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Knowledge of linear algebra, probability and statistics.	
Course Contents	Approximate syllabus: - quantum states, unitary operations, and measurements; - universal gate sets; - entanglement, Bell test; - basic quantum communication protocols; - basic algorithms and quantum algorithmic techniques; - basic quantum error correction; - simple physical implementations of qubits.	
Study Goals	Motivation: Quantum information is the future of computing and communication. Quantum computers offer exponential speedup over any classical computer. Similarly, quantum communication offers many advantages, including the ability to create secure encryption keys where security rests only on the laws of nature. Synopsis: This class will teach you the fundamental principles of quantum information. You will learn essential concepts that distinguish quantum from classical devices. You will learn about quantum bits and the quantum operations and measurements that can be performed on them. You will learn the basic techniques used in quantum algorithms, and examine basic examples of such algorithms. You will also take the first step in understanding how a quantum bit can be physically implemented. Aim: To learn the fundamental concepts underlying quantum computation and communication systems.	
Education Method	3 hours of lecture, 1 hour tutorial per week.	
Literature and Study Materials	The main reference textbook for the course is Nielsen and Chuang, Quantum Computation and Information, Cambridge University Press.	
Assessment	30% homework assignments, 10% in class quiz, 60% final exam. A minimum grade of 5.0 (unrounded) for the final exam is required to pass the course.	
Permitted Materials during Tests		
Continuing Courses	This course forms part of the curriculum on Quantum Technologies offered at TU Delft, which at present consists of: AP3421 Fundamentals of quantum information AP3421-PR Quantum Information Project CS4090 Quantum communication and cryptography AP3432 Quantum Hardware 1 - Theoretical Concepts AP3442 Quantum Hardware 2 - Experimental State of the Art EE4575 Electronics for quantum computation	

CS4035	Cyber Data Analytics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	The course provides theoretical and practical background for applying data analytics in the field of cyber security. Cyber data analytics is a huge field with a great diversity of techniques and applications. The course is centered on a selection of seven such techniques: learning from imbalanced data; behavioral profiling and anomaly detection; sequential data mining; data stream mining; learning from software data; adversarial machine learning; and privacy-aware data mining; Anomaly detection is one of the main topics in cyber security. Specific difficulties that the student will learn to handle are the huge amounts of data and the large number of false positives. Behavioral profiling applies to both people and software processes. Different techniques will be taught to construct profiles from software logs. While building such profiles, care should be taken to not infringe upon the privacy of individuals the data is collected from. Finally, attackers will modify their behavior in order to avoid being detected, a cyber data analytics engineer tries to make their models/profiles robust against such modifications.	
Study Goals	The student will be able to: Apply machine learning to real data Understand and modify machine learning algorithms Learn models from time series Detect anomalies in multidimensional time-series Use distributed processing to speed up machine learning Learn models from data streams with limited memory Learn sequential models Use machine learning for fingerprinting and profiling Preserve the privacy of data owners while learning models Learn robust models that can detect evasive attackers Use machine learning to detect fraud, attacks, and botnets	
Education Method	There will be two lectures for each of the seven topics, and 3 large lab assignments on fraud detection, anomaly detection, and behavioral profiling, and 1 smaller lab on adversarial robustness. There is no exam. Teams of two students will work on these assignments which contain both individual and collaborative components. Deadlines are strict as peer-review will be used to both learn of other possible solutions, provide feedback, and get initial estimates on the obtained grade.	
Assessment	3 large lab assignments on fraud detection (30%), anomaly detection (30%), and behavioral profiling (30%), and 1 smaller lab on adversarial robustness (10%).	

CS4090	Quantum Communication and Cryptography	5
Responsible Instructor	Dr. S.D.C. Wehner	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Expected prior knowledge	Linear Algebra, Probability & Statistics, Q101 (Fundamentals of quantum information)	
Course Contents	This class will introduce you to the fascinating field of quantum communication! We will look at the state of the art of quantum networks, and explore techniques for building quantum repeaters that promise to deliver qubits over long distances. We also briefly look at one of the most famous application of quantum cryptography, quantum key distribution. Caution: 1. This class requires you to take "Fundamentals of Quantum Information" in Quarter 1 2. The focus of this class is presently on quantum communication, and we will only briefly look at quantum cryptography. As such, this class is not held in flipped classroom style in conjunction with edX QuCryptoX as in previous years.	
Study Goals	The student will acquire: A good understanding of the fundamental concepts of quantum communication Insight into the differences between classical and quantum communication and cryptography Skill set required to follow the remainder of the quantum curriculum (Q301 Quantum hardware and Q401 Quantum electronics)	
Education Method	Lectures and tutorials. If remote classes continue this fall, then recorded lectures and live discussion session	
Literature and Study Materials	Primary: Slides Review Articles Auxilliary: Nielsen and Chuang Quantum computation and information, Cambridge University Press.	
Assessment	Homework (70%) and Final Project (30%) disclaimer: information may change depending on the developments around the coronavirus.	
Tags	Abstract Adventurous Algorithmics Challenging Group Dynamics/Project Organisation Information & Communication Integrated Intensive Involved Lineair Algebra Mathematics Physics Quantum Signals Technology Telecommunication	

CS4110	Artificial Intelligence for Software Testing and Reverse Engineering	5
Responsible Instructor	Ir. S.E. Verwer	
Responsible Instructor	Dr. A. Panichella	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	Software is one of the most complex artifacts of mankind has ever created, but complexity is the enemy of correctness. Modern software testing and validation tools use a multitude of techniques geared toward correct computer code, most of these are base on artificial intelligence. In this course, we study these techniques in details, specifically we will understand and implement: Execution monitoring and taint analysis Branch distance computation Hill-climbing and genetic algorithms Concrete and symbolic (concolic) execution Active state machine learning Genetic programming The goal is to better understand and test software using artificial intelligence. Using the taught techniques you will be able to automatically: Discover which code is reachable Find (security) bugs in software Write tests that cover all reachable code Reverse engineer a code's functionality Patch code to remove bugs and failing tests	
Study Goals	The student will: Understand modern AI techniques for software testing. Be able to implement several such techniques from scratch: - smart fuzzing (probing software with input to find crashes/bugs), - symbolic execution (using logic to construct inputs that trigger specific code branches), - fault localization (given that a program fails, find the line of code responsible for the failure), and - automated program repair (using a patch library and genetic programming to improve code) Be able to apply this technology to locate bugs in real-world software implementations.	
Education Method	The main part of the course will consist of 3 lab assignments covering the theory (fuzzing&tainting, symbolic execution, automated program repair), and one lab assignment for the application to real software. The students will implement the taught techniques from scratch in the first 3 assignments, which will be scored with a pass/fail. All three assignments need to be passed to complete the course. The final lab will contain a recap from the first three assignments and an application of a state-of-the-art tool on real software. The final lab will be graded and be the final course grade. There will be instruction sessions where students can work on their assignment and ask the teachers for assistance.	
Assessment	First three lab assignments (pass/fail). Final lab (100%).	
Tags	Artificial intelligence Software	

CS4120	Seminar Science and Methods in Cyber security	5
Responsible Instructor	Dr. S. Picek	
Instructor	Dr. M.P.M. Franssen	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	This seminar course Cyber Security covers the following topics: (i) an introduction to the philosophy of (classical and design) science, (ii) the art of writing a scientific research proposal, (iii) an overview of useful and relevant scientific methods, (iv) introduction to scientific writing (of a paper and of a MSc thesis).	
Study Goals	1. Getting a basic knowledge and understanding of what science entails and how scientific knowledge is being created 2. Getting knowledge and understanding of relevant scientific methods applicable in the field of Cyber Security 3. Getting knowledge, understanding and skills for writing a research proposal related to the creation of a MSc thesis 4. Getting knowledge and understanding on how to execute a scientific article and MSc thesis 5. Getting knowledge and understanding of how to execute a literature review.	
Education Method	Lecturers supported by the execution of mostly individual assignments. Attendance of participants in this course is mandatory.	
Assessment	Final grade will be based on a weighted average of various scores including (i) presence and level of participation (10%), (ii) quality of the research proposal to be written and presented (60%), (iii) grades for assignments (paper evaluation, paper rewrite, essay questions/written exam) (30%). No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Tags	Research Methods	

CS4150	Systems Security	5
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/0/6	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	Language based Security (CS4105) and Security and Cryptography (IN4191) and Network Security (ET4397IN) and a Bachelor level Operating Systems course. The topics below should be covered to a good bachelor level. To allow students to assess their level of knowledge, they can have a short oral interview.	
Course Contents	IoT security, Hardware, Countermeasures, Covert channels, Secure System Engineering	
Study Goals	The student will acquire: An appreciation of the security architecture of computer systems Detailed knowledge of the security of a specific operating system Skills in exploiting vulnerabilities of computer systems Skills in developing counter measures against exploits	
Education Method	2 hours per week lectures, 4 hours per week lab	
Assessment	50% lab work (with automated testing for scalability) and 50% written open book examination. disclaimer: information may change depending on the developments around the coronavirus.	

CS4160	Blockchain Engineering	5
Responsible Instructor	Dr.ir. J.A. Pouwelse	
Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	In this course you will learn all aspects of blockchain technology, including tamper-proof data structures, digital identities, transitive trust, fault tolerance, distributed consensus, smart contracts and applications. Ledgers and blockchains are an emerging technology with the potential to radically improve financial transactions, supply-chain flows, transactions in general, and distributed databases. The first three weeks of the course will provide a fast-paced introduction to Bitcoin, Ethereum, and TrustChain developed by TUDelft itself. The main component in this course is a team-based complex engineering project. This course is designed for computer scientists to understand blockchain technology and to produce significant hands-on experience. To provide a deep understanding of blockchain technology and understand why it is special you need to experience first-hand how it operates at a detailed technical level. Students design, implement, and test their own independent project in teams of 3-5 students. Students can choose from a pool of possible project ideas. This course requires you to like software engineering. Topics covered: -Blockchain basics and evolution Bitcoin 1st generation, smart contract generation, future 3rd generation (trust or trust in math) -identity and transitive trust Authentication and security primitives, tamper-proof identities, trust models, MITM attacks, Sybil attacks, and TrustChain by TUDelft -Consensus models Proof-of-work, permissioned, Proof-of-stake, Corda no-global-consensus, TUDelft bottom-up fast consensus model -Smart Contract pro/con encrypted data, Bitcoin scripts, Etherium execution model, Hyperledger + Docker issues, Corda Jar file approach, Tezos difficult to use, powerful technology, vision of the future: trusted verified execution -Markets and exchanges Disruption by open markets, winner-takes-all, and multi-sided market platforms, Uber, Airbnb, 22 years of eBay, Silk Road, honesty among drug dealers, the role of trust in markets, P2P exchange markets -Decentralized Autonomous Organization, novel method to collaborate and organise any economic activity Within this 2021 edition "the Delft DAO" will be prominently featured. TUDelft achieved a world-first in DAO research. We devised a full end-to-end proof-of-principle of a DAO which is capable of 0) near unbounded scalability 1) controlling money 2) democratic decision making and 3) continuous sustained self-evolution. This course provides you with the knowledge to work with this advanced technology. After this course you will have a firm grasp on the current operational blockchain-based systems, realistic view of real-world applications that may be built on top of ledger technology. You will be able to reason and discuss the open challenges and questions that still need to be resolved. This course is a key course for distributed systems students.	
Study Goals	After this course students are able to design and engineer complex blockchain-based systems. Students are able to describe blockchain technology, the various consensus model, smart contracts, markets, and relation to existing database technology. Student are able to setup a new architecture for blockchain applications.	
Education Method	This course consists of four 2-hour lectures. Each lecture is followed by a 4-hour homework period in the same week focused on understanding the background material. In week 1 you will form teams and initiate work on your blockchain engineering project. A list of projects to select from will be provided at the start of this course.	
Literature and Study Materials	Online course textbook: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Narayanan, Bonneau, Felten, Miller and Goldfeder.	
Prerequisites	It is highly recommended to follow this course (see remarks): Security and Cryptography (Q1) Distributed Algorithms (Q2)	
Assessment	The final grade reflects the quality of your work and team cooperation. This course has a minimal amount of formalities. You will write down your project results in a single-page report, IEEE style. You will be graded on your open source efforts located on Github and single-page report. Your grade will be expressed on a scale of 0 to 10. (resits or repair options are not offered for this course) Covid-19 disclaimer: the assessment and course format could be altered at any time !!!	
Remarks	This class has a limited capacity (50). If there is a larger number of enrollments than the capacity of the class, students will be assigned to their preferred blockchain engineering project based on their background, engineering experience level, and match to the course goals. Students who followed Security and Cryptography (Q1) and are also enrolled in Distributed Algorithms (Q2) will have priority for placement. Mathematics students are exempts from this, if they can show some minimal software development experience (e.g. Github profile). Finally, students with a Grade Point Average of 8.0 or higher are eligible for the challenging scientific projects, resulting in a research paper. These project receive intense guidance, but have no capacity limits.	

CS4185	Capstone Cyber Security	5
Responsible Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	2/1/1/1	
Education Period	1 2 3 4	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	Please note that the course has 3 parts. All parts must be done within 12 months from the start of part 1 and in the following order: 1. Q1 & Q2: Capstone Social Skills https://www.4tu.nl/cybsec/en/course-program/cps/ 2. Q3 & Q4: Capstone Entrepreneurial skills https://www.4tu.nl/cybsec/en/course-program/cpe/ 3. Q5*: Capstone Business skills https://www.4tu.nl/cybsec/en/course-program/cpb/ *) Be aware that if you start this course in 2020-2021 (part 1 and 2), then part 3 will start at the beginning of the next academic year 2021-2022. Attendance is mandatory and there is no (chance of) reparation if you miss any session. NOTE: the study guide information may change depending on the developments around the coronavirus	
Study Goals	see the links provided	
Education Method	see the links provided	
Assessment	see the links provided disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	15 participants max, 4TU.CybSec students have priority.	
Co-Instructor	Dr.ing. V.E. Scholten	

CS4265	Computer and Network Security: Advanced Topics	5
Responsible Instructor	Prof.dr. M. Conti	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	*DISCLAIMER: study guide information may change depending on the developments around the coronavirus.* The course takes the form of seminars based on a selection of scientific papers (that either have had a strong impact on security today, or explore novel ideas that may be important in the future). The list of topics can be found in the brightspace Topics and Papers module. For each topic there is a primary paper, and possibly other additional papers. All the students are required to read all primary papers and be able to competently discuss the material in class. Each student will be responsible for presenting one lecture (based on one of the primary paper including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion in the class. 48 hours before each lecture each student must upload on a shared repository at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). This is intended to be an interactive class: class participation is strongly recommended (and will play a role in the grading criteria). Sleeping during the class is optional, but not recommended.	
Study Goals	This course is about learning to study, analize, do and criticize research in cybersecurity. This will be done by being exposed to actual research topics and scientific papers and discussing things together.	
Education Method	Studying, presenting and discussing recent research results in Computer and Network Security.	
Assessment	Presentation + Class Discussion + Written Report + Oral Exam (please refer to the Judgement field for more details) disclaimer: information may change depending on the developments around the coronavirus.	
Elective	Yes	
Judgement	The final grade will be made up of four components: 25% the presentation done by the student during the course: each student will be responsible for presenting one topic (based on the corresponding primary paper, including as much relevant related work as necessary to distill the work presented in the paper). The speaker will have a finite time (20 minutes) to present the papers. The presentation will be followed by 10 minutes of interactive discussion (Q&A) in the class. This component is based on following criteria: (15%) Layout and Graphics (30%) Content (20%) Organization (20%) Presentation (15%) Q&A. 25% for the active participation in Q&A sessions during the course: 48 hours before each lecture each student must submit (via email, to both the lecturer and the teaching assistants) at least two thought-provoking questions for each one of the main papers covered in the lecture. These questions should critically evaluate the papers (e.g., questioning the assumptions, criticize the methodology, compare with other solutions, propose alternative solutions, etc.). The students should actively participate in the discussion of the topics in the 10 minutes Q&A session for each presented topic. 25% for content and quality of the final essay: At the end of the course, each student must write a 5-page long essay about one of the topics that has been discussed in class, or another topic agreed with the lecturer. The topic and the structure of the essay must be agreed with the lecturer. The essay might include some implementation prototype or experiments/simulations to evaluate/support the claim in the paper (in case this is a significant part of the essay, two students can agree with the lecturer to work together). If the student cannot attend the lectures, an alternative work (e.g. a longer essay) must be agreed with the lecturer. 25% for the oral presentation of the essay: during the oral exam, the student is asked to give a 15-minute presentation to the lecturer and the teaching assistants about the essay (presenting with slides is highly recommended). During the oral presentation, students can also be asked questions about other topics of the course. This component is based on following criteria: (30%) Style (20%) Originality (50%) Organization (clarity in your argumentation, coherence between assumptions and conclusions, logical organization, evidence to support claims)	
Co-Instructor	Ir. S.E. Verwer	

CS4280	Language-Based Software Security	5
Responsible Instructor	Dr. J.G.H. Cockx	
Contact Hours / Week x/x/x/x	0/0/0/4	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	This course has no formal prerequisites. However, for the homework assignments you will have to implement several program analysis techniques using the Scala programming language. If you have not used Scala before, you are thus expected to learn the basics of the language through self-study.	
Course Contents	Security vulnerabilities often arise due to programming errors in the source code of an application. Recent programming errors with severe security implications include Heartbleed (buffer over-read), Shellshock (code injection), and goto-fail (ill-formated code). Rather than hunt for individual vulnerabilities in programs, a more structural approach to improve security is to improve the programming language. This is the goal of language-based security: to rule out whole classes of potential security vulnerabilities in one go. This course studies various security properties and program analysis techniques for enforcing these properties at the level of the programming language to improve software security. In particular, we will study the following properties: - Memory safety: prevent buffer overflows and overreads - Type safety: prevent undefined behaviour - Information flow control: prevent data leaks and code injection attacks We will study techniques to address these problems at the language level through dynamic analysis, static analysis, and language design. To facilitate a precise study and comparison, we will define the above techniques formally in class. To facilitate student experimentation and exploration of trade-offs, students will implement the above techniques in homework assignments.	
Study Goals	After taking this course, students should be able to: 1. Describe the nature and causes of security vulnerabilities in software systems, and give concrete examples of how these security vulnerabilities can be exploited. 2. Explain the properties that can be enforced at the level of the programming language to rule out security vulnerabilities, such as memory safety, type safety, and non-interference. 3. Formally define the semantics of a simple programming language. 4. Formally define dynamic and static analysis techniques for enforcing these security properties. 5. Implement these techniques for a small programming language. 6. Discuss and evaluate the importance of soundness and precision of a given program analysis. 7. Contrast programming languages based on the set of countermeasures they provide, and give an appropriate recommendation for a specific application. 8. Analyse and apply results from scientific literature in the area of language based security.	
Education Method	The course work consists of the following activities: 1 or 2 instruction sessions per week. Weekly homework assignments consisting of theoretical questions, programming assignments, and reading assignments	
Assessment	The assessment for this course consists of two parts: The weekly homework assignments will test your ability to design an implement (variants of) the techniques discussed in the lectures (study goals 3-5). This counts for 40% of the total grade. The final written or oral exam will test your theoretical understanding of the security vulnerabilities and their countermeasures discussed in class (study goals 1-2) and your ability to discuss and contrast the different aspects of these techniques (study goals 6-8). This counts for 60% of the total grade. To pass the course, each of these grades (homework assignments and final exam) should be 5.0 or higher, and the final grade should be 5.8 or higher (and will be rounded to the nearest half grade point). disclaimer: information may change depending on the developments around the coronavirus.	
Co-Instructor	E. Visser	

CS4430	Network Security	5
Responsible Instructor	Dr.ing. A. Zarras	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	0/0/4/0	
Education Period	3	
Start Education	3	
Exam Period	3 4	
Course Language	English	
Course Contents	The course provides an overview of the most important concepts, methods, and best practices in computer and network security. In this course, students will obtain the knowledge and hands-on experience to secure networking and communication systems. The course's primary focus will be on technologies, protocols, attacks, and defenses. More precisely, starting from a review of common vulnerabilities and attack scenarios, the course will discuss the fundamentals of security engineering and their application in system design, review tools and methods to assess and test communication infrastructure from a security perspective. As a result, students will gain theoretical knowledge and hands-on experience in network attacks and defense methods. Knowledge activation and the transfer from conceptual understanding towards practical experience will be further facilitated by students implementing their own attack or defense tools on selected topics, as well as conducting measurements on the effectiveness of attack and defense schemes.	
Study Goals	See course contents.	
Education Method	Lectures, Labs, and Project.	
Assessment	Assignments and Project.	

IN4191	Security and Cryptography	5
Responsible Instructor	Z. Erkin	
Instructor	Dr. K. Liang	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Required for	UT-201500042 Privacy Enhancing Technologies (Q4)	
Expected prior knowledge	Basic understanding on the following is suggested. -Probability and statistics -Programming skills	
Course Contents	Motivation: Computers are now found in every layer of society, and information is being communicated and processed automatically on a large scale. Examples include medical and financial files, automatic banking, video-phones, pay-tv, teleshopping and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy. Synopsis: Security and cryptography are essential components of any digital system. In this course, the fundamentals of secure data storage and transportation of information are described. In particular, classical (e.g. Caesar, Vigenere) and modern encryption schemes (RSA, DES, AES, Elliptic curves) are described along with their mathematical background such as number theory. Methods for authentication, data integrity and digital signatures are discussed in detail, as these are the main components of many security architectures. The course also investigates more advanced topics such as zero-knowledge proofs and secret sharing schemes. Aim: It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security and privacy, as well as is familiar with present applications. Learning outcomes: The goal is to make students familiar with the basic concepts applied cryptography, including classical cryptography and modern secret key and public key cryptography. In particular, the students will acquire A sound understanding of the notion of security An understanding of the confidentiality, integrity and authenticity needs of the society Understand the role of cryptographic primitives including the differences between symmetric and asymmetric cryptography, the role of hash functions, digital signatures and PKI Understand the advanced topics in cryptography needed for the modern society with untrustworthy entities Among others things, the following topics are covered: -Classical systems -Information theoretic security -Definition of Security notions -Symmetric encryption (e.g. DES, AES) -Asymmetric encryption (RSA, Elliptic Curves) -Hash functions -Random number generation -Key Management -Digital Signatures, -*Secret Sharing. (if time permits) -*Zero Knowledge proofs (if time permits)	
Study Goals	It is the aim that at the end of the course one has a survey of the state of the art of both cryptographic algorithms and protocols for security, as well as is familiar with present applications. Through assignments, students are expected to have the chance to work on the topics covered in the lectures.	
Education Method	Lectures, assignments and weekly exercises. Attention: This course requires full effort of 140 hours. Even more, if you lack the background (probability and modular arithmetic) Planned Workload: Lectures: 28 x 45minutes sessions, total 22 hours Practice session: 7 x 90 min. total 12 hours Assignments: 3 x 20 hour, total 60 hours Weekly study: 7 x 4 hours, total 28 hours Exam preparation: 20 hours Exam: 3 hours	
Literature and Study Materials	Cryptography made simple, Nigel P. Smart, 2nd Edition, Springer, 2016 (PDF Available Online) Handouts of lectures	
Assessment	Written exam(70%) + mandatory assignments (30%). Passing grade for the written exam and assignments is required. Please refer to CS regulations for further details. There is NO reparation for assignments. Points from previous years cannot be transferred. NOTE: the study guide information may change depending on the developments around the COVID-19 pandemic	
Exam Hours	In case of in person examination at campus: The exam is closed book. A cheat sheet of size A4, hand written is allowed for the written exam. Name and student number has to be present on each side. In case of remote exam from home: open book: textbook, slides and self-made notes only. randomised/customised exam.	

Permitted Materials during Tests	Only non-scientific calculators.

IN4253ET	**"Hacking Lab"-Applied Security Analysis**	**5**
Responsible Instructor	Dr. S. Picek	
Contact Hours / Week x/x/x/x	0/0/2/0	
Education Period	3	
Start Education	3	
Exam Period	none	
Course Language	English	
Expected prior knowledge	Necessary background differs per student project, see first lecture or contact instructors for details	
Course Contents	The security of computer and telecommunication systems is becoming an increasing concern. In this course, we will review the current state of the art on security research and gain practical experience in assessing the security and vulnerabilities of communication systems. Engineers are typically taught to focus on performance, correctness, scalability, and maintainability when building communication and information processing systems. However, an additional set of design principles are required to achieve security. In this course, we discuss security principles, common pitfalls and vulnerabilities. The weekly lectures provide an introduction into security research, with a focus on real-world security, privacy-enhancing technology and common security pitfalls. Each student participates in a "Hack Project", with a group of one to four students. Students can select between a wide range of available Hack Project outlines within the first week. The goal may be to evaluate the security of a real-world IT system, developing a proof-of-concept exposing a vulnerability or focussed on preserving privacy in a post-Snowden world. Students may propose their own Hack Project based on their background knowledge and skills. Such Hack Projects need to be approved and shaped together with the instructor. Example of possible outlined hardware-oriented projects are: development of a wifi tracker, programing an FPGA system to break passwords, assess the security of RFID cards, or to transparently intercept Ethernet traffic. Concrete software projects are: hacking Bitcoin, improving the TOR anonymity protocol and create Android-based tools for human rights activists in Iran, Egypt and Russia, reprogramming neural networks attacks. Each Hack Project is documented with a written report. This can be in the form of a 6-8 page IEEE-style scientific article or a traditional more lengthy report. All results, experiences and findings are presented to the entire class in the last week of the course. Hack Projects also report their progress several times during the course, after the weekly lectures.	
Study Goals	After this course, the student will have a thorough knowledge of security in real-world systems, and will be able to explore the literature on this topic independently. The student will be aware of the poor state of security in real-world computer systems. The student can explain the common pitfalls, why these known failures still occur and reasons behind the poor state of security in general.	
Education Method	Lectures, student presentations, written final report and active participation. Attendance and active participation during lectures is mandatory. This sadly means telelecturing is not possible.	
Literature and Study Materials	Customize literature lists and study materials are provided per project topic	
Assessment	The final class grade is composed of several partial grades. Partial grades are given for the written Hack Project report (60%), final presentation of result (10%), presentation of ongoing project progress (20%), participation in discussions, overall quality of the practical work and class attendance (10%). Students are required to obtain a passing grade on all partial grades. Attendance to lectures is mandatory. No final written exam. No resit will be offered of any practical work. If a student passes only part of the course, all parts need to be retaken. disclaimer: information may change depending on the developments around the coronavirus.	
maximum aantal deelnemers	If there is an unexpected high demand for this course, then enrollment will be based on past performance in relevant courses.	

IN5000	Final Project	45
Responsible Instructor	Dr.ir. A.R. Bidarra	
Responsible Instructor	Dr.ir. W.P. Brinkman	
Responsible Instructor	Dr. K.A. Hildebrandt	
Responsible Instructor	Dr. J.C. van Gemert	
Responsible Instructor	M. Finavaro Aniche	
Responsible Instructor	Dr. C. Lofi	
Responsible Instructor	Prof.dr. J.S. Rellermeyer	
Contact Hours / Week x/x/x/x	x/x/x/x	
Education Period	1 2 3 4 Summer Holidays	
Exam Period	none	
Course Language	English	
Course Contents	IN5000 Final Project is the final part of the Master's degree programme. During this project, you will be required to demonstrate your ability to solve a research or engineering problem. The project must be carried out using the techniques of project management. You will begin by making a project plan in cooperation with your Masters thesis advisor. Several aspects of the project are defined within the plan, including the assignment, the frequency of interaction with the advisors, the milestones of the project and the resources and facilities offered by the faculty. You will be required to adhere to your plan throughout the project. It is obviously possible to adjust your plan under certain circumstances and after discussion with your daily supervisor. At the end of the project, you will submit your Masters thesis, which must be written in English, and make an oral presentation of your work to the Thesis Committee. The Thesis Committee will announce the final mark, which is based on the project performance, the thesis, the presentation and the subsequent discussion. More information about the graduation process: https://www.tudelft.nl/en/student/faculties/eemcs-student-portal/education/graduation-policy-msc/	
Study Goals	1. The student is able to design a research project: - The student is able to use, explain and justify adequate research and design methodologies; - The student is able to apply theory to the performed project; - The student is able to use techniques for interpretation and verification and bases his/her conclusions on results; - The student is able to do reliable work with scientific significance. 2. The student is able to execute a research project: - The student has a critical attitude towards his/her own results, literature and specialists; - The student makes an original contribution to the project; - The student takes initiative (together with the supervisor) to give his/her own input within the research project; - The student interacts sufficiently with peers and superiors; - The student is able to make and execute a project plan. 3. The student is able to write a research report: - The student is able to write a research report that shows sufficient coherence of content; - The students is able to structure the research report and sufficiently present the content (text and figures); - The student expresses argumentation using correct spelling and grammar; 4. The student is able to present and defend the research project: - The student is able to present the content using sufficient detail to support conclusions; - The student is able to logical structure the presentation and use visual aids; - The student is able to adequately formulate and express himself/herself as well as sufficiently address the audience; - The student is able to argument and answer the questions asked by the committee.	
Education Method	Project	
Prerequisites	Before starting the project, students must have completed at least 60 EC of the Master's degree programme and be in possession of a Thesis Entrance Permit (TEP). To be able to get a TEP, the individual exam program (IEP) should have been approved by the Board of Examiners. Note: In some cases, the thesis supervisor may impose additional conditions for starting this project.	
Assessment	The thesis committee assesses the thesis and the defense on the following criteria: - quality of work: novelty, volume, grasp, methodology, publishable; - personal performance: autonomy, planning, creativity, attitude; - quality of thesis report: clarity, organisation, argumentation; - oral presentation and defense: clarity, focus, relevance, discussion. More information on thesis grading: https://www.tudelft.nl/en/student/faculties/eemcs-student-portal/education/graduation-policy-msc/assessment/ The voting members of the thesis committee determine the final grade. The grade should reflect a weighted average of the four scores above, but need not to be an exact arithmetical mean. The final mark starts from 5 up to and 10. Marks ending in .5 may also be used. If the student shows excellence (is nominated for a 10) the chair of the thesis committee should consult the chair of the Board of Examiners, at least five working days in advance of the defense. The chair may advice to add an extra member to the thesis committee. The motivation for the grade at each of the four criteria as listed above is summarized on a form and signed by the chairman of the thesis committee. The candidate is given a short account of the assessment, either in private or in front of the audience. disclaimer: information may change depending on the developments around the coronavirus.	

TPM020A	Economics of Cybersecurity	5
Module Manager	S.E. Parkin	
Instructor	S.E. Parkin	
Co-responsible for assignments	Prof.dr. M.J.G. van Eeten	
Contact Hours / Week x/x/x/x	2x2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	With a significant increase in high-profile data breaches and cybersecurity threats in the last couple years, it is critical for businesses to learn about the costs and investment decisions around securing their online systems. While many businesses think of cybersecurity as a technical problem, this course broadens that view and shows that security failures are caused as often by bad business decisions and incentive systems as by bad technical design. This course provides an introduction to the field of the economics behind cybersecurity. It will provide you with the economic concepts, measurement approaches to make better security decisions, while helping you to understand the forces that shape the security decisions of other businesses, products and services.	
Study Goals	The student will: -Gain a sound understanding of the economics of cybersecurity as a systems discipline, from security policies (modelling what ought to be protected) to mechanisms (how to implement the protection goals). -Obtain foundational skills in identifying and assessing data on complex decisions around information security issues. -Gain insights into the design of effective policies to enhance and maintain cyber security must take into account a complex set of incentives facing not only the providers and users of the Internet and computer software, but also those of potential attackers. -Learn to apply economic analysis and analytic approaches to the open issues and pending activities in cybersecurity.	
Education Method	Structured lectures, background reading, group discussions, and regular non-graded quizzes.	
Assessment	Group report and presentation on assigned reading material (20%); the final assignment is an essay (80%).	

TPM025A	User-Centred Security	5
Module Manager	S.E. Parkin	
Instructor	S.E. Parkin	
Responsible for assignments	S.E. Parkin	
Co-responsible for assignments	Prof.dr. M.J.G. van Eeten	
Contact Hours / Week x/x/x/x	0/0/0/x	
Education Period	4	
Start Education	4	
Exam Period	4	
Course Language	English	
Course Contents	Cybersecurity attacks on organizations and services increasingly target the people who are involved, such as employees in companies or home computer users. Solutions to address this problem need to be feasible for individuals, alongside other user priorities (such as completing paid work). Without consideration of user skills or needs, solutions such as security training or browser warning pop-ups add effort and burden to the user, and encourage less secure behaviours as workarounds (such as writing down difficult-to-remember passwords to ensure system access). In this course, students will learn about the user perspective of security technologies. This will leverage key human factors concepts, around security usability and its connections to decisions in policy, planning, and technology investment. This will include how to assess a security solution from the perspective of different kinds of technology users and their tasks. By assessing the strengths and weaknesses of particular security mechanisms for users in practice (policies, training, monitoring, etc.), security implementation and management decisions can be made which better fit the context in which mechanisms are used. This can ensure long-term security which better matches the requirements of a particular user organisation or community.	
Study Goals	The student will: -Gain a sound understanding of security usability as a discipline, from assessing the context of use alongside primary tasks, to identifying the time and effort costs to users. -Obtain foundational skills in matching security technologies and processes to user abilities, motivations, and perceptions of security-related technologies. This will include examination of authentication and identity technologies, employee security training, trust in online contexts, privacy-related evaluations of personal data disclosure, etc. -Gain insights into the design of effective security technologies and their deployment from a human-centred perspective, to inform interface and policy design and ensure compliance with security expectations. This will include management of security behaviour change activities, and identifying how to position expert support to aid users, e.g., effective communication of policy, use of persuasive design in security.	
Education Method	Structured lectures, background reading, problem-driven group discussions	
Assessment	Individual assignment on assigned reading material (20%); a final individual assignment is an essay (80%)	
Elective	Yes	

TPM027A	Cyber Risk Management	5
Module Manager	Prof.dr.ir. P.H.A.J.M. van Gelder	
Instructor	K. Labunets	
Instructor	Prof.dr.ir. P.H.A.J.M. van Gelder	
Responsible for assignments	Prof.dr.ir. P.H.A.J.M. van Gelder	
Co-responsible for assignments	M. Daneva	
Contact Hours / Week x/x/x/x	4/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	MOTIVATION: The challenge of selecting the optimal technical, organisational, legal, and other preventive and repressive measures to reduce cyber risks to acceptable levels can only be understood in the context of the application of Cyber Risk Management. Risk Management is about analysing the relationships between threats, incidents and risks (here in the complex world of cyberspace), based on which an adequate set of countermeasures can be designed. SYNOPSIS: Risk (= the potential lo losing something of value) can manifest itself in cyberspace in all kinds of ways: values at stake are financial wealth, health, physical condition (of people, materials, goods, infrastructures, etc.), well-being, reputation, privacy, trust, etc. Based on a conceptualisation of cyberspace and its various sub-domains (discussed in the project week of year 1), we introduce risk assessment approaches, both of qualitative and quantitative manner, illustrated with case studies. CONTENT: Cyberspace Risk concepts Probabilistic and adversarial risk System and attacker models Requirements and policies Risk assessment methods Qualitative and quantitative methods Measuring security and effectiveness	
Study Goals	After this course, students are able to: Understand and apply risk concepts to (possible) incidents in cyberspace Understand the theoretical principles of cyber risk management Explain the role of adversarial risk models, system models, and attacker profiles in cyber risk assessment Analyze and apply state-of-the-art cyber risk assessment methodologies to (complex, multi-step) cyber incidents Analyze the (expected) effect of measures that help to prevent the occurrence of cyber incidents Justify investments in cyber security	
Education Method	LECTURES: There are two lecture slots per week EXAM: There is a written exam at the end of the course. LANGUAGE: The course is taught in English. LECTURERS: Prof. dr. ir. Pieter v. Gelder (TUD/TPM) and guest lecturers.	
Assessment	Grading will be based on the written exam at the end of the course (100%).	
Elective	Yes	
Targetgroup	MSc	
Category	MSc level	

TPM030A	Introduction to Cloud as Infrastructure: The effects of the new business of computing on practice	5
Module Manager	Dr. F.S. Gürses	
Instructor	Dr. F.S. Gürses	
Co-responsible for assignments	Prof.dr. M.J.G. van Eeten	
Contact Hours / Week x/x/x/x	0/2x2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	Why is there a rush to AI? What is the role of cloud computing in the multi-trillion dollar valuation of tech companies? Are mobile phones really personal devices? How has the way software is produced today changed the theory and practice of computing? And, what are the implications of the technical breakthroughs pushed by big tech companies (Google, Apple, Facebook, Amazon and Microsoft (GAFAM)) for our societies? These are some of the questions we will ask during this course. In the hands of GAFAM, software production has gone through fundamental changes that impact how computing and our society is organized. While Information and Communication Technologies (ICTs) have always been important for the functioning of public and private organizations, the turn to agile production with services based on current computational infrastructures (e.g., clouds, personal devices, (sensor) networks) organizes computing in a way that increases dependencies on the tech giants. These dependencies bring about significant changes that affect people, institutions and our common infrastructures (e.g., health, education, transportation, energy). These changes require a deeper understanding and a broader reflection as we implement digital services in all aspects of life using computational infrastructures dominated by GAFAM. This course will walk you through prevalent changes in the production of software and the business of computing, and their impact on our societies. Throughout the course, you will find out about shifts in production and how (geo)political, economic and technological factors brought these into being. We will do so through case studies of turning points in the production of software and the associated computational infrastructures. We will study these turning points concretely in the context of GAFAMs. Examples of such turning points include the turn from personal computers to (mobile) devices attached to the clouds, waterfall methods to agile/lean methodologies, monolith architectures to (micro)services, instruction-based programming to AI, and the move from general-purpose to specialized chips. During the course, you will learn how these new forms of computing and software production aim to reconfigure organizations and how this is tied to the ever-growing business of computing. As importantly, we will reflect on what this new form of production means for developers, engineers and (computer) scientists and the potentials and constraints it imposes on our ability to design systems. To be able to draw a comparison, we will look at how each shift in software production changes associated privacy concerns and how big tech companies have proposed to address them through privacy-by-design.	
Study Goals	Hence, in this course you will discover how the way we produce software impacts how society is organized by reaching the following learning-objectives: Demonstrate a basic understanding of the history of computing through exemplary shifts in software production Compare the different roles big technology companies (Google, Apple, Facebook, Amazon, Microsoft shorthanded as GAFAM) play in providing current day computational infrastructures Illustrate how software production has become more dependent on computational infrastructures Reflect on the impact of new forms of software production on the capabilities of engineers, developers and scientists in designing and deploying systems Analyze and assess different privacy concerns, and corresponding privacy technologies, that arise with shifts in software production Explain how the rise of ML/AI technologies are related to computational infrastructures. Speculate and predict possible futures of software production in the hands of GAFAM	
Education Method	Lectures, work groups, case studies, presentations, self-study, writing.	
Assessment	10% engagement 50% 5 assignments 40 % oral exam The total of the assignments as well as the oral exam must be graded with at least 5.0. The final grade consists of the average of the three parts (engagement, assignments and exam), and the rounded grade must be at least 6.0 (5.75 rounded up).	
Elective	Yes	
Category	MSc level	

UT-191612680	Computer Ethics	5
Responsible Instructor	mr. J.M. Kooijman	
Contact Hours / Week x/x/x/x	0/4/0/0	
Education Period	2	
Start Education	2	
Exam Period	none	
Course Language	English	
Course Contents	https://www.4tu.nl/cybsec/en/course-program/coe/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=191612680&collegejaar=2020&taal=en	
Assessment	One written take-home exam at the end of the quartile, which counts 40% of the grade, and one group assignment (essay), which counts for the remaining 60%. The resit policy is that students are allowed to resit either the assignment or the exam once, irrespective of their initial grade.	
Enrolment / Application	Only for students enrolled in the 4TU cybersecurity master track of computer science. Requirements: - An UT student guest account is required. See https://www.utwente.nl/onderwijs/student-services/procedures-services/aanmelding-inschrijving/bijvakker-of-kies-op-maat/#registratie-bijvakker-of-kies-op-maat - The exam will be held at the TU Delft; therefore students must register here: https://www.tudelft.nl/studenten/onderwijs/tentamens/aanmelden-voor-tentamens/	

UT-192110940	Secure Data Management	5
Responsible Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	2/0/0/0+ assignments	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/sdm/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=192110940&collegejaar=2020&taal=en	

UT-201100022	Cyber Crime Science	5
Responsible Instructor	Dr. R.S. van Wegberg	
Contact Hours / Week x/x/x/x	0,0,2,1+project	
Education Period	3 4	
Start Education	3	
Exam Period	none	
Course Language	English	
Course Contents	0,0,2,1+project	

UT-201500038	E-Law	5
Responsible Instructor	mr. J.M. Kooijman	
Module Manager	mr.dr L.C.P. Broos	
Instructor	mr. J.M. Kooijman	
Contact Hours / Week x/x/x/x	0/0/0/2	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Course Contents	See the website https://www.4tu.nl/cybsec/en/course-program/	
Enrolment / Application	Only for students enrolled in the 4TU cybersecurity master track of computer science. Requirements: - A UT student guest account is required. See https://www.utwente.nl/onderwijs/student-services/procedures-services/aanmelding-inschrijving/bijvakker-of-kies-op-maat/#registratie-bijvakker-of-kies-op-maat - The exam will be held at the TU Delft; therefore students must register here: https://www.tudelft.nl/studenten/onderwijs/tentamens/aanmelden-voor-tentamens/	

UT-201500039	Security Verification	5
Contact Hours / Week x/x/x/x	2/0/0/0+ assignments	
Education Period	1	
Start Education	1	
Exam Period	none	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/sev/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=201500039&collegejaar=2020&taal=en	

UT-201500040	Introduction to Biometrics	5
Responsible Instructor	Ir. S.E. Verwer	
Contact Hours / Week x/x/x/x	0/2/0/0+ project	
Education Period	2	
Start Education	2	
Exam Period	none	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/bio/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=201500040&collegejaar=2020&taal=en	

UT-201500041	Cyber Security Management	5
Responsible Instructor	Prof.dr.ir. P.H.A.J.M. van Gelder	
Module Manager	Prof.dr.ir. P.H.A.J.M. van Gelder	
Contact Hours / Week x/x/x/x	0/2/0/0	
Education Period	2	
Start Education	2	
Exam Period	2 3	
Course Language	English	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/csm/ https://osiris.utwente.nl/student/OnderwijsCatalogusSelect.do?selectie=cursus&cursus=201500041&collegejaar=2019&taal=en	
Enrolment / Application	Only for students enrolled in the 4TU cybersecurity master track of computer science. Requirements: - A UT student guest account is required. See https://www.utwente.nl/onderwijs/student-services/procedures-services/aanmelding-inschrijving/bijvakker-of-kies-op-maat/#registratie-bijvakker-of-kies-op-maat - The exam will be held at the TU Delft; therefore students must register here: https://www.tudelft.nl/studenten/onderwijs/tentamens/aanmelden-voor-tentamens/	

UT-201500042	Privacy Enhancing Technologies	5
Responsible Instructor	Z. Erkin	
Contact Hours / Week x/x/x/x	0/0/0/2+ assignments	
Education Period	4	
Start Education	4	
Exam Period	4 5	
Course Language	English	
Expected prior knowledge	IN4191 Security and Cryptography Course	
Course Contents	http://www.3tu.nl/cybsec/en/course-program/pet/ anonymous communication; identity management; anonymous credentials; anonymity systems; mix networks; onion routing; database privacy; k-anonymity; differential privacy; other probabilistic approaches; private data processing; secure multiparty computation; (fully/somewhat) homomorphic encryption; garbled circuits; secret sharing; privacy-preserving clustering; private recommender systems; private smart metering; privacy-preserving biometrics.	
Study Goals	Concepts like the Internet of Things or Big Data inherently utilize massive amounts of data containing private information collected and stored by websites, sensors, monitoring systems, auditing systems, and so on. Examples include electronic records in health care systems and location information in ubiquitous computing applications. But how can we protect the privacy of participating users while at the same time enable effective sharing and utilization of the distributed data? There are several dimensions in the area of privacy, ranging from technical and juridical to societal and economical. While we will touch upon all these different aspects in the course, we will focus on the technical dimension. We will explore potential techniques for building new platforms, services, and tools that protect users' privacy. The study of promising component technologies ranging from advances in anonymous communication and identity management to theoretic tools like differential privacy and cryptography will be the core of this course. Learning objectives: Good understanding of privacy in the Internet of Things Ability to analyse and evaluate anonymity mechanisms, both for anonymous communication and for database privacy Ability to apply and analyse the concept of secure multiparty computation to protect privacy in different application domains Gain hands-on experience with different privacy-enhancing technologies	
Education Method	Lectures (joint lecturing), lectures slides and articles.	
Literature and Study Materials	Lecture Slides, essential reading material provided with the lectures (articles and tutorials)	
Assessment	70% written, closed book, exam and 30% assignments. Passing grade for the written exam is required. In case of remote exam from home: open book exam, with random oral checks. NOTE: the study guide information may change depending on the developments around the coronavirus	

UT-202000026	Secure Cloud Computing	5
Responsible Instructor	F. Hahn	
Course Coordinator	Dr. S. Picek	
Contact Hours / Week x/x/x/x	2/0/0/0	
Education Period	1	
Start Education	1	
Exam Period	1 2	
Course Language	English	
Expected prior knowledge	Students should have solid foundational knowledge of cryptography, as for instance covered in the course Security and Cryptography (201500027), and should have previously gained some working experience with a common programming language such as C or C++ and Python.	
Course Contents	Platform-as-a-Service; Virtualization; Sandboxing; Key Management; Database-as-a-Service; Searchable Encryption; Attacks on Searchable Encryption; Oblivious RAM; Private Information Retrieval; Functional Encryption; Secure Multiparty Computation; Homomorphic Encryption; Intel SGX; Verifiable Computation; Machine-Learning-as-a-Service; Model Inversion Attack	
Study Goals	Cloud computing allows clients to rent major parts of their computing infrastructure instead of owning and maintaining large data centers. Due to virtualization techniques, this approach is scalable and gives flexibility in the used computational resources. Resources can be adapted as required, while the underlying hardware is provided and maintained by the cloud service provider. However, outsourcing vital business data and delegating business critical tasks requires trust in the cloud service provider. It is not surprising that a lack of such required trust is still one of the main obstacles for the full adoption of cloud computing. This course covers security mechanisms specifically suitable for cloud computing. After a brief introduction to cloud computing, we discuss security mechanisms currently offered by big players such as Amazon and Microsoft and identify potential shortcomings thereof. The major content of this course presents recent research directions regarding data security in cloud computing scenarios. Among others, we will discuss constructions for outsourced encrypted databases and potential attacks on such systems, verifiable computation, hardware-aided security and privacy issues of outsourced data analytics. Note that Stjepan Picek is TU Delft contact person, but not in charge of the course content	
Education Method	2x 45 min. lectures per week + (practical) assignment	
Literature and Study Materials	Various papers from the literature	
Prerequisites	Security and Cryptography	
Assessment	(Closed book) written exam (70%); three (practical) assignments (30%) disclaimer: information may change depending on the developments around the coronavirus.	

Prof.dr.ir. K.I. Aardal

Unit	Elektrotechn., Wisk. & Inform.
Department	Discrete Wiskunde en Optimalisering
Telephone	+31 15 27 85093
Room	36.HB 04.030

Dr. T.E.P.M.F. Abeel

Unit	Elektrotechn., Wisk. & Inform.
Department	Pattern Recognition and Bioinformatics
Telephone	+31 15 27 85114
Room	28.6.W840

Dr. B.P. Ahrens

Department	Programming Languages
Telephone	+31 15 27 89972

S. Baars

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 81220
Room	31.C0.150

Dr. H.J.E. Beaumont

Department	BN/Bertus Beaumont Lab
Telephone	+31 15 27 88647
Room	58.F1.430

Unit	Technische Natuurwetenschappen
Department	BN/Bertus Beaumont Lab
Telephone	+31 15 27 88647
Room	58.F1.430

Prof.dr.ir. N. Bharosa

Department	Informatie en Communicatie Technologie
Telephone	+31 15 27 82402

Dr.ir. A.R. Bidarra

Unit	Elektrotechn., Wisk. & Inform.
Department	Computer Graphics and Visualisation
Telephone	+31 15 27 84564
Room	28.5.W820

M. Bliekendaal

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 86359
Room	31.C0.050

Drs. W.J. Blokzijl

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 86414
Room	31.c0.120

J. Borregaard

Department	QN/Borregaard groep
Room	22.F 033c

Prof.dr. P.A.N. Bosman

Unit	Elektrotechn., Wisk. & Inform.
Department	Algorithmics

Dr. E.F.M. van Boven

Unit	Elektrotechn., Wisk. & Inform.
Department	Network Architectures and Services
Telephone	+31 15 27 87853
Room	36.HB 09.240

Dr. S.J. van Boxtel

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 81893
Room	31.c0.090

Prof.dr.ir. A. Bozzon

Department	Web Information Systems
Telephone	+31 15 27 87822
Room	32.B-3-370

Unit	Elektrotechn., Wisk. & Inform.
Department	Sustainable Design Engineering
Telephone	+31 15 27 87822
Room	32.B-3-370

Prof.dr. F.M. Brazier

Department	Systeemkunde
Telephone	+31 15 27 87529

Unit	Techniek, Bestuur & Management
Department	Systeemkunde
Telephone	+31 15 27 87529

Dr.ir. W.P. Brinkman

Unit	Elektrotechn., Wisk. & Inform.
Department	Interactive Intelligence
Telephone	+31 15 27 83534
Room	28.4.W800

mr.dr L.C.P. Broos

F. Broz

Department	Interactive Intelligence
Room	28.4.W560

Prof.mr.dr. J.A. de Bruijn

Unit	Techniek, Bestuur & Management
Department	Organisatie en Governance
Telephone	+31 15 27 87169

Dr. M.L.C. de Bruijne

Department	Organisatie en Governance
Telephone	+31 15 27 82323

Unit	Techniek, Bestuur & Management
Department	Organisatie en Governance
Telephone	+31 15 27 82323
Room	31.B2.080

Dr. M. Bruijnes

Ir. B.J.E. de Bruin

Department	Cognitive Robotics

Department	Cognitive Robotics

Dr. J.W. Böhmer

Department	Sequential Decision Making
Telephone	+31 15 27 86110

P.S. Cesar Garcia

Department	Multimedia Computing
Room	28.5.E160

Unit	Elektrotechn., Wisk. & Inform.
Department	Multimedia Computing
Room	28.5.E160

Dr. S.S. Chakraborty

Department	Programming Languages
Telephone	+31 15 27 85867
Room	28.3.E260

Dr. Y. Chen

Unit	Elektrotechn., Wisk. & Inform.
Department	Data-Intensive Systems
Telephone	+31 15 27 89025

Dr. J.G.H. Cockx

Department	Programming Languages
Room	28.3.E340

Prof.dr. M. Conti

Department	Cyber Security
Telephone	+31 15 27 87647
Room	29.02.470

M. Daneva

Dr. E. Demirovi

Department	Algorithmics
Telephone	+31 15 27 87177
Room	28.4.E400

Prof.dr. A. van Deursen

Unit	Elektrotechn., Wisk. & Inform.
Department	Software Engineering
Telephone	+31 15 27 82486
Room	28.1.W600

Dr. L. di DiCarlo

Unit	QuTech
Department	QCD/DiCarlo Lab
Telephone	+31 15 27 86097
Room	22.B 267

Unit	Technische Natuurwetenschappen
Department	QN/DiCarlo Lab
Telephone	+31 15 27 86097
Room	22.B 267

Dr. Y. Ding

Unit	Techniek, Bestuur & Management
Department	Informatie en Communicatie Technologie
Telephone	+31 15 27 85659
Room	31.B3.260

Prof.dr. M.J.G. van Eeten

Unit	Techniek, Bestuur & Management
Department	Organisatie en Governance
Telephone	+31 15 27 87050
Room	31.b2.160

Prof.dr. E. Eisemann

Unit	Elektrotechn., Wisk. & Inform.
Department	Computer Graphics and Visualisation
Telephone	+31 15 27 82528
Room	28.5.W860

Dr. D. Elkouss Coronas

Unit	Elektrotechn., Wisk. & Inform.
Department	Quantum Computer Science
Telephone	+31 15 27 88029
Room	22.B 204

Prof.dr.ir. D.H.J. Epema

Department	Data-Intensive Systems
Telephone	+31 15 27 83853
Room	28.1.E060

Unit	Elektrotechn., Wisk. & Inform.
Department	Distributed Systems
Telephone	+31 15 27 83853
Room	28.1.E060

Z. Erkin

Unit	Elektrotechn., Wisk. & Inform.
Department	Cyber Security
Telephone	+31 15 27 86283
Room	29.02.250

Dr. S. Feld

Department	QCD/Feld Group
Telephone	+31 15 27 82462

Department	Quantum Circuit Architectures and Technology
Telephone	+31 15 27 82462
Room	36.HB 11.060

M. Finavaro Aniche

Department	Software Engineering
Telephone	+31 15 27 84211

Unit	Elektrotechn., Wisk. & Inform.
Department	Software Engineering
Telephone	+31 15 27 84211
Room	28.4.W660

Dr. M.P.M. Franssen

Department	Ethiek & Filosofie van Techniek
Telephone	+31 15 27 85795
Room	31.b4.300

Unit	Techniek, Bestuur & Management
Department	Ethiek & Filosofie van Techniek
Telephone	+31 15 27 85795
Room	31.b4.300

U.K. Gadiraju

Department	Web Information Systems
Telephone	+31 15 27 87271
Room	28.3.W740

Prof.dr.ir. P.H.A.J.M. van Gelder

Unit Techniek, Bestuur & Management
Department Safety and Security Science

Telephone +31 15 27 86544
Room 31.c1.170

Dr. J.C. van Gemert

Unit Elektrotechn., Wisk. & Inform.
Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 88434
Room 28.6.E340

A. Glasbergen-Plas

Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 89798

Dr.ir. D.M. Groenewegen

Unit Elektrotechn., Wisk. & Inform.
Department Computer Science & Engineering-Teaching Team

Room 28.3.E240

Dr. F.S. Gürses

Department Organisatie en Governance

Telephone +31 15 27 89863

F. Hahn

Prof.dr. A. Hanjalic

Unit Elektrotechn., Wisk. & Inform.
Department Intelligent Systems

Telephone +31 15 27 83084
Room 28.5.E360

Dr. C. Hauff

Unit Elektrotechn., Wisk. & Inform.
Department Web Information Systems

Telephone +31 15 27 87107
Room 36.HB 08.110

J. He

Department Interactive Intelligence

Telephone +31 15 27 87598

Department Sequential Decision Making

Telephone +31 15 27 87598

Department Interactive Intelligence

Telephone +31 15 27 87598

Prof.dr.ir. J. Hellendoorn

Department College van Bestuur

Telephone +31 15 27 89007
Room 34.F-2-440

Unit Mech, Maritime & Materials Eng
Department Cognitive Robotics

Telephone +31 15 27 89007
Room 34.F-2-440

Dr.ir. R.C. Hendriks

Unit	Elektrotechn., Wisk. & Inform.
Department	Signal Processing Systems
Telephone	+31 15 27 87191
Room	36.HB 17.050

Dr. K.A. Hildebrandt

Unit	Elektrotechn., Wisk. & Inform.
Department	Computer Graphics and Visualisation
Telephone	+31 15 27 81445
Room	28.5.W620

F. Hirschhorn Zonana

Drs. G. Hoezen

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 82915
Room	31.C0.110

Prof.dr.ir. G.J.P.M. Houben

Unit	Elektrotechn., Wisk. & Inform.
Department	Web Information Systems
Telephone	+31 15 27 85953
Room	28.3.W840

Y. Huang

Unit	Techniek, Bestuur & Management
Department	Systeemkunde
Telephone	+31 15 27 83822

Dr. H.S. Hung

Unit	Elektrotechn., Wisk. & Inform.
Department	Pattern Recognition and Bioinformatics
Telephone	+31 15 27 85697

T. Höllt

Department	Computer Graphics and Visualisation
Telephone	+31 15 27 87506

Dr. G. Iosifidis

Department	Networked Systems

Prof.dr.ir. M.F.W.H.A. Janssen

Unit	Techniek, Bestuur & Management
Department	Engineering, Systems and Services
Telephone	+31 15 27 81140
Room	31.b3.150

S.F. Johnson

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 82793

Prof.dr. C.M. Jonker

Unit	Elektrotechn., Wisk. & Inform.
Department	Interactive Intelligence
Telephone	+31 15 27 81315
Room	28.6.W600

G. Joseph

Department Signal Processing Systems

Telephone +31 15 27 87456
Room 36.HB 17.040

Drs. A.E. Kam

Unit Techniek, Bestuur & Management
Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 89728
Room 31.c0.100

Dr. A. Katsifodimos

Unit Elektrotechn., Wisk. & Inform.
Department Data-Intensive Systems

Telephone +31 15 27 87247
Room 28.4.E080

mr. J.M. Kooijman

Unit Techniek, Bestuur & Management
Department Organisatie en Governance

Telephone +31 15 27 81173
Room 31.b2.220

Dr.ir. J.H. Krijthe

Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 88433
Room 28.5.E820

Prof.dr.ir. F.A. Kuipers

Unit Elektrotechn., Wisk. & Inform.
Department Networked Systems

Telephone +31 15 27 81347
Room 28.2.E160

Drs. B.M.D. van der Laaken

Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 81160

Unit Techniek, Bestuur & Management
Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 81160
Room 31.c0.050

K. Labunets

Department Organisatie en Governance

Telephone +31 15 27 81052

Dr. P.H.G. van Langen

Unit Techniek, Bestuur & Management
Department Systeemkunde

Telephone +31 15 27 83429
Room 31.b1.170

Prof.dr. K.G. Langendoen

Unit Elektrotechn., Wisk. & Inform.
Department Embedded Systems

Telephone +31 15 27 87666

Drs. D.W. Laponder

Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 89799

Drs. D.W. Laponder

Dr. M. Leijten

Unit Techniek, Bestuur & Management
Department Organisatie en Governance

Telephone +31 15 27 83561

Prof.dr.ir. G.J.T. Leus

Unit Elektrotechn., Wisk. & Inform.
Department Signal Processing Systems

Telephone +31 15 27 84327
Room 36.HB 17.280

Dr. K. Liang

Department Cyber Security

Telephone +31 15 27 87656

Dr.ir. C.C.S. Liem

Unit Elektrotechn., Wisk. & Inform.
Department Multimedia Computing

Telephone +31 15 27 82188
Room 28.5.E300

Prof.dr.ir. H.X. Lin

Unit Elektrotechn., Wisk. & Inform.
Department Mathematical Physics

Telephone +31 15 27 87229
Room 36.HB 05.060

Drs. K.I. van der Linden

Unit Techniek, Bestuur & Management
Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 86384
Room 31.c0.140

Dr. R. Litjens

Unit Elektrotechn., Wisk. & Inform.
Department Network Architectures and Services

Telephone +31 15 27 86622

Dr. C. Lofi

Unit Elektrotechn., Wisk. & Inform.
Department Web Information Systems

Room 28.3.W780

M. Loog

Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 89395
Room 28.6.E100

Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 89395
Room 28.6.E100

Unit Elektrotechn., Wisk. & Inform.
Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 89395
Room 28.6.E100

M. Looij

Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 89821
Room 31.C0.080

Dr. R. Marroquim

Unit Elektrotechn., Wisk. & Inform.
Department Computer Graphics and Visualisation

Telephone +31 15 27 81988
Room 28.5.W800

Dr. J.A. Martinez Castaneda

Unit Elektrotechn., Wisk. & Inform.
Department Multimedia Computing

Telephone +31 15 27 85445
Room 28.5.E060

L. Meester

Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 89812

Dr.ir. F.H. van der Meulen

Department Statistics

Telephone +31 15 27 84517
Room 36.HB 06.040

Unit Elektrotechn., Wisk. & Inform.
Department Statistics

Telephone +31 15 27 84517
Room 36.HB 06.040

Dr.ir. E. Minkman

Department Organisatie en Governance

L. Miranda da Cruz

Department Software Engineering

Telephone +31 15 27 87113

Dr. P.K. Murukannaiah

Department Interactive Intelligence

Telephone +31 15 27 82291
Room 28.4.W900

Prof.dr. M.A. Neerincx

Unit Elektrotechn., Wisk. & Inform.
Department Interactive Intelligence

Telephone +31 15 27 87106
Room 28.6.W680

Dr. C.R.M.M. Oertel Genannt Bierbach

Department Interactive Intelligence

Telephone +31 15 27 87503

Dr. F.A. Oliehoek

Unit Elektrotechn., Wisk. & Inform.
Department Sequential Decision Making

Telephone +31 15 27 87494
Room 28.4.W600

Dr. A. Panichella

Unit Elektrotechn., Wisk. & Inform.
Department Software Engineering

Telephone +31 15 27 89306
Room 28.1.W800

S.E. Parkin

Department Organisatie en Governance

Dr. P. Pawelczak

Unit Elektrotechn., Wisk. & Inform.
Department Embedded Systems

Telephone +31 15 27 87491
Room 28.2.W680

Dr. S. Picek

Department Cyber Security

Telephone +31 15 27 89319

Dr. J.S. de Pinho Gonçalves

Unit Elektrotechn., Wisk. & Inform.
Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 83418
Room 28.6.W900

S. Pintea

Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 88037

Drs. P.C. Post

Unit Techniek, Bestuur & Management
Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 83175
Room 31.c0.040

C.B. Poulsen

Department Software Technology

Unit Elektrotechn., Wisk. & Inform.
Department Programming Languages

Room 28.3.E400

Dr.ir. J.A. Pouwelse

Unit Elektrotechn., Wisk. & Inform.
Department Data-Intensive Systems

Telephone +31 15 27 82539
Room 36.HB 07.290

Dr.ing. S. Proksch

Department Software Engineering

Telephone +31 15 27 87215

Prof.dr.ir. M.J.T. Reinders

Unit Elektrotechn., Wisk. & Inform.
Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 86424
Room 28.6.W600

Prof.dr. J.S. Rellermeyer

Department Data-Intensive Systems

Telephone +31 15 27 84571

Department	Data-Intensive Systems
Telephone	+31 15 27 84571
Room	28.3.E360

Prof.dr.ir. G.A. de Reuver

Unit	Techniek, Bestuur & Management
Department	Informatie en Communicatie Technologie
Telephone	+31 15 27 81920
Room	31.B3.210

G.A. De Reuver

Prof.dr. B. Rieger

Unit	Technische Natuurwetenschappen
Department	ImPhys/Rieger group
Telephone	+31 15 27 88574
Room	22.F 266

Dr. S. Roos

Department	Data-Intensive Systems
Telephone	+31 15 27 87001

Unit	Elektrotechn., Wisk. & Inform.
Department	Dataintensive Systems
Telephone	+31 15 27 87001

Dr.ing. V.E. Scholten

Unit	Techniek, Bestuur & Management
Department	Delft Centre for Entrepreneurship
Telephone	+31 15 27 89596
Room	31.C2.070

L.C. Schroten

Unit	Techniek, Bestuur & Management
Department	Instituut voor Talen en Academische Vaardigheden
Telephone	+31 15 27 87301
Room	31.C0.120

Dr. F. Sebastiano

Department	QCD/Sebastiano Lab
Telephone	+31 15 27 83342

Unit	Elektrotechn., Wisk. & Inform.
Department	Quantum Circuit Architectures and Technology
Telephone	+31 15 27 83342
Room	36.HB 11.080

Prof.dr.ir. D.M. van Solingen

Department	Software Engineering
Room	28.4.W800

Department	Software Engineering
Room	28.4.W800

Unit	Elektrotechn., Wisk. & Inform.
Department	Software Engineering
Room	28.4.W800

Prof.dr. M.T.J. Spaan

Unit	Elektrotechn., Wisk. & Inform.
Department	Sequential Decision Making
Telephone	+31 15 27 81102
Room	28.4.E340

Prof. M.M. Specht

Unit Elektrotechn., Wisk. & Inform.
Department Web Information Systems

Telephone +31 15 27 87457
Room 28.2.W600

Prof.dr.ir. D. Spinellis

Department Software Engineering

Room 28.1.W900

Ir. B.M. Steenhuisen

Unit Techniek, Bestuur & Management
Department Organisatie en Governance

Telephone +31 15 27 81329
Room 31.b2.060

Drs. M.A. Swennen

Unit Techniek, Bestuur & Management
Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 82912
Room 31.C0.160

Dr. D.M.J. Tax

Unit Elektrotechn., Wisk. & Inform.
Department Pattern Recognition and Bioinformatics

Telephone +31 15 27 84232
Room 28.5.W860

Prof.dr. B.M. Terhal

Unit QuTech
Department Discrete Wiskunde en Optimalisering

Telephone +31 15 27 81940

Unit Elektrotechn., Wisk. & Inform.
Department Discrete Wiskunde en Optimalisering

Telephone +31 15 27 81940

M.L. Tielman

Unit Elektrotechn., Wisk. & Inform.
Department Interactive Intelligence

Telephone +31 15 27 87516
Room 28.4.W580

Prof.dr. W. Tittel

Unit QuTech
Department QID/Tittel Lab

Telephone +31 15 27 87393
Room 22.B 003

Unit Elektrotechn., Wisk. & Inform.
Department Quantum Communications Lab

Telephone +31 15 27 87393
Room 22.B 003

J. Urbano Merino

Unit Elektrotechn., Wisk. & Inform.
Department Multimedia Computing

Telephone +31 15 27 86176
Room 28.5.E280

Prof.dr.ir. P.F.A. Van Mieghem

Unit Elektrotechn., Wisk. & Inform.
Department Network Architectures and Services

Telephone +31 15 27 82397
Room 36.HB 09.270

Prof.dr.ir. L.M.K. Vandersypen

Unit QuTech
Department QCD/Vandersypen Lab

Telephone +31 15 27 82469
Room 22.F 216

Unit Technische Natuurwetenschappen
Department QN/Vandersypen Lab

Telephone +31 15 27 82469
Room 22.F 216

Prof.dr.ir. A.J. van der Veen

Unit Elektrotechn., Wisk. & Inform.
Department Signal Processing Systems

Telephone +31 15 27 86240
Room 36.HB 17.030

Dr. R.R. Venkatesha Prasad

Unit Elektrotechn., Wisk. & Inform.
Department Networked Systems

Telephone +31 15 27 86272

Ir. S.E. Verwer

Unit Elektrotechn., Wisk. & Inform.
Department Algorithmics

Telephone +31 15 27 88435
Room 28.6.E100

E. Visser

Unit Elektrotechn., Wisk. & Inform.
Department Programming Languages

Telephone +31 15 27 87088
Room 36.

Dr. F.M. Vos

Unit Technische Natuurwetenschappen
Department ImPhys/Vos group

Telephone +31 15 27 87133
Room 22.D 205

M.J.Y. Wackers

Unit Techniek, Bestuur & Management
Department Instituut voor Talen en Academische Vaardigheden

Telephone +31 15 27 85634
Room 31.C0.060

B. Wagner

Department Organisatie en Governance

Room 31.B2.180

H. Wang

Unit Elektrotechn., Wisk. & Inform.
Department Multimedia Computing

Telephone +31 15 27 88847
Room 28.5.E260

Q. Wang

Department Embedded Systems

Room 28.2.W700

Dr.ir. J.H. Weber

Unit	Elektrotechn., Wisk. & Inform.
Department	Discrete Wiskunde en Optimalisering
Telephone	+31 15 27 81698
Room	36.HB 04.040

Prof.dr. M.M. de Weerdt

Unit	Elektrotechn., Wisk. & Inform.
Department	Algorithmics
Telephone	+31 15 27 84516
Room	28.4.E100

Dr. R.S. van Wegberg

Department	Organisatie en Governance
Telephone	+31 15 27 81320
Room	31.B2.210

Dr. S.D.C. Wehner

Unit	QuTech
Department	QID/Wehner Group
Telephone	+31 15 27 87746

Unit	Elektrotechn., Wisk. & Inform.
Department	Quantum Computer Science
Telephone	+31 15 27 87746

Ir. K.F. Wullaert

Department	Programming Languages
Telephone	+31 15 27 87945

J. Yang

Department	Web Information Systems
Room	28.3.W760

J. Yang

Dr. N. Yorke-Smith

Unit	Elektrotechn., Wisk. & Inform.
Department	Algorithmics
Telephone	+31 15 27 83895
Room	28.4.E220

Prof.dr. A.E. Zaidman

Unit	Elektrotechn., Wisk. & Inform.
Department	Software Technology
Telephone	+31 15 27 84385
Room	28.1.W860

Dr.ing. A. Zarras

Department	Cyber Security

M.A. Zuñiga Zamalloa

Unit	Elektrotechn., Wisk. & Inform.
Department	Networked Systems
Telephone	+31 15 27 82538
Room	28.2.E080

Dr. B. Özkan

Department	Software Engineering
Room	28.1.W660