# Program overview

**Year**: 2018/2019  
**Organization**: Electrical Engineering, Mathematics and Computer Science  
**Education**: Master Computer Science

<table>
<thead>
<tr>
<th>Code</th>
<th>Omschrijving</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Track Data Science &amp; Technology 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Common Core DST 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS4035</td>
<td>Cyber Data Analytics</td>
<td>5</td>
</tr>
<tr>
<td>CS4065</td>
<td>Multimedia Search and Recommendation</td>
<td>5</td>
</tr>
<tr>
<td>IN4010(-12)</td>
<td>Artificial Intelligence Techniques</td>
<td>6</td>
</tr>
<tr>
<td>IN4085</td>
<td>Pattern Recognition</td>
<td>6</td>
</tr>
<tr>
<td>IN4086-14</td>
<td>Data Visualization</td>
<td>6</td>
</tr>
<tr>
<td>IN4252</td>
<td>Web Science &amp; Engineering</td>
<td>5</td>
</tr>
<tr>
<td>IN4301</td>
<td>Advanced Algorithms</td>
<td>5</td>
</tr>
<tr>
<td>IN4315</td>
<td>Software Architecture</td>
<td>5</td>
</tr>
<tr>
<td>IN4391</td>
<td>Distributed Systems</td>
<td>5</td>
</tr>
<tr>
<td><strong>Specialisatievakken start kwartaal</strong></td>
<td><strong>Specialisation courses start first period 2018</strong></td>
<td></td>
</tr>
<tr>
<td><strong>1 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP3421</td>
<td>Fundamentals of Quantum Information</td>
<td>4</td>
</tr>
<tr>
<td>CS4010</td>
<td>Algorithms for Planning and Scheduling</td>
<td>5</td>
</tr>
<tr>
<td>CS4070</td>
<td>Multivariate Data Analysis</td>
<td>5</td>
</tr>
<tr>
<td>CS4130</td>
<td>Seminar Programming Languages</td>
<td>5</td>
</tr>
<tr>
<td>CS4165</td>
<td>Seminar Social Signal Processing</td>
<td>5</td>
</tr>
<tr>
<td>CS4200-B</td>
<td>Compiler Construction Project</td>
<td>5</td>
</tr>
<tr>
<td>EE4C06</td>
<td>Networking</td>
<td>5</td>
</tr>
<tr>
<td>ET4388</td>
<td>Ad-hoc Networks</td>
<td>5</td>
</tr>
<tr>
<td>IN4010(-12)</td>
<td>Artificial Intelligence Techniques</td>
<td>6</td>
</tr>
<tr>
<td>IN4049TU</td>
<td>Introduction to High Performance Computing</td>
<td>6</td>
</tr>
<tr>
<td>IN4085</td>
<td>Pattern Recognition</td>
<td>6</td>
</tr>
<tr>
<td>IN4191</td>
<td>Security and Cryptography</td>
<td>5</td>
</tr>
<tr>
<td>IN4252</td>
<td>Web Science &amp; Engineering</td>
<td>5</td>
</tr>
<tr>
<td>IN4301</td>
<td>Advanced Algorithms</td>
<td>5</td>
</tr>
<tr>
<td>IN4309</td>
<td>Random Signal Processing</td>
<td>5</td>
</tr>
<tr>
<td>IN4310</td>
<td>Seminar Computer Graphics</td>
<td>5</td>
</tr>
<tr>
<td>IN4314</td>
<td>Seminar Selected Topics in Multimedia Computing</td>
<td>5</td>
</tr>
<tr>
<td>IN4326</td>
<td>Seminar Web Information Systems</td>
<td>5</td>
</tr>
<tr>
<td>IN4334</td>
<td>Analytics and Machine Learning for Software Engineering</td>
<td>5</td>
</tr>
<tr>
<td>IN4387</td>
<td>System Validation</td>
<td>5</td>
</tr>
<tr>
<td>IN4392</td>
<td>Cloud Computing</td>
<td>5</td>
</tr>
<tr>
<td>IN4398</td>
<td>Advanced Practical IoT and Seminar</td>
<td>5</td>
</tr>
<tr>
<td>IN4402</td>
<td>Computer and Network Security: advanced topics</td>
<td>5</td>
</tr>
<tr>
<td><strong>Specialisatievakken start kwartaal</strong></td>
<td><strong>Specialisation courses start second period 2018</strong></td>
<td></td>
</tr>
<tr>
<td><strong>2 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS4015</td>
<td>Behaviour Change Support Systems</td>
<td>5</td>
</tr>
<tr>
<td>CS4055</td>
<td>High Performance Data Networking</td>
<td>5</td>
</tr>
<tr>
<td>CS4090</td>
<td>Quantum Communication and Cryptography</td>
<td>5</td>
</tr>
<tr>
<td>CS4106</td>
<td>Dynamic and Static Program Analysis for Software Security</td>
<td>5</td>
</tr>
<tr>
<td>CS4160</td>
<td>Blockchain Engineering</td>
<td>5</td>
</tr>
<tr>
<td>CS4190</td>
<td>Psychology of Programming</td>
<td>5</td>
</tr>
<tr>
<td>IN4086-14</td>
<td>Data Visualization</td>
<td>6</td>
</tr>
<tr>
<td>IN4150</td>
<td>Distributed Algorithms</td>
<td>6</td>
</tr>
<tr>
<td>IN4302TU</td>
<td>Building Serious Games</td>
<td>5</td>
</tr>
<tr>
<td>IN4341</td>
<td>Performance Analysis</td>
<td>5</td>
</tr>
<tr>
<td><strong>Specialisatievakken start kwartaal</strong></td>
<td><strong>Specialisation courses start third period 2018</strong></td>
<td></td>
</tr>
<tr>
<td><strong>3 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AP3132</td>
<td>Advanced Digital Image Processing</td>
<td>6</td>
</tr>
<tr>
<td>CS4110</td>
<td>Artificial Intelligence for Software Testing and Reverse Engineering</td>
<td>5</td>
</tr>
<tr>
<td>CS4125</td>
<td>Seminar Research Methodology for Data Science</td>
<td>5</td>
</tr>
<tr>
<td>CS4135</td>
<td>Software Verification</td>
<td>5</td>
</tr>
<tr>
<td>CS4180</td>
<td>Deep Learning</td>
<td>6</td>
</tr>
<tr>
<td>CS4195</td>
<td>Modeling and Data Analysis in Complex Networks</td>
<td>5</td>
</tr>
<tr>
<td>CS4210-A</td>
<td>Algorithms for Intelligent Decision Making</td>
<td>5</td>
</tr>
<tr>
<td>EE4560</td>
<td>Information Theory</td>
<td>5</td>
</tr>
<tr>
<td>ET4394</td>
<td>Wireless IoT and Local Area Networks</td>
<td>5</td>
</tr>
<tr>
<td>ET4397IN</td>
<td>Network Security</td>
<td>5</td>
</tr>
<tr>
<td>IN4152</td>
<td>3D Computer Graphics and Animation</td>
<td>5</td>
</tr>
<tr>
<td>IN4179</td>
<td>Intelligent User Experience Engineering</td>
<td>6</td>
</tr>
</tbody>
</table>
### Informatievakken start kwartaal 4 2018

<table>
<thead>
<tr>
<th>Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN423ET</td>
<td>&quot;Hacking Lab&quot;-Applied Security Analysis</td>
<td>5</td>
</tr>
<tr>
<td>IN4315</td>
<td>Software Architecture</td>
<td>5</td>
</tr>
<tr>
<td>IN4320</td>
<td>Machine learning</td>
<td>5</td>
</tr>
<tr>
<td>IN4325</td>
<td>Information Retrieval</td>
<td>5</td>
</tr>
<tr>
<td>IN4343</td>
<td>Real-time Systems</td>
<td>5</td>
</tr>
<tr>
<td>IN4391</td>
<td>Distributed Systems</td>
<td>5</td>
</tr>
<tr>
<td>IN4393</td>
<td>Computer Vision</td>
<td>5</td>
</tr>
</tbody>
</table>

### Specialisation courses start fourth period 2018

<table>
<thead>
<tr>
<th>Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS4035</td>
<td>Cyber Data Analytics</td>
<td>5</td>
</tr>
<tr>
<td>CS4065</td>
<td>Multimedia Search and Recommendation</td>
<td>5</td>
</tr>
<tr>
<td>CS4140ES</td>
<td>Embedded Systems Laboratory</td>
<td>5</td>
</tr>
<tr>
<td>CS4145</td>
<td>Crowd Computing</td>
<td>5</td>
</tr>
<tr>
<td>CS4205</td>
<td>Evolutionary Algorithms</td>
<td>5</td>
</tr>
<tr>
<td>CS4210-B</td>
<td>Intelligent Decision Making Project</td>
<td>5</td>
</tr>
<tr>
<td>ET4030</td>
<td>Error Correcting Codes</td>
<td>4</td>
</tr>
<tr>
<td>ET4285</td>
<td>Measuring and Simulating the Internet</td>
<td>4</td>
</tr>
<tr>
<td>IN4182</td>
<td>Digital Audio and Speech Processing</td>
<td>6</td>
</tr>
<tr>
<td>IN4185</td>
<td>Globally Distributed Software Engineering</td>
<td>5</td>
</tr>
<tr>
<td>IN4254</td>
<td>Smart Phone Sensing</td>
<td>5</td>
</tr>
<tr>
<td>IN4255</td>
<td>Geometric Data Processing</td>
<td>5</td>
</tr>
<tr>
<td>IN4306</td>
<td>Literature Survey</td>
<td>10</td>
</tr>
<tr>
<td>IN4331</td>
<td>Web-scale Data Management</td>
<td>5</td>
</tr>
<tr>
<td>IN4333</td>
<td>Language Engineering Project</td>
<td>5</td>
</tr>
<tr>
<td>Track Data Science &amp; Technology 2018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Program Coordinator</strong></td>
<td>Drs. M.A. van Loo</td>
<td></td>
</tr>
<tr>
<td><strong>MSc Coordinator</strong></td>
<td>Dr. K.A. Hildebrandt</td>
<td></td>
</tr>
<tr>
<td><strong>MSc Coordinator</strong></td>
<td>Dr. C. Lofi</td>
<td></td>
</tr>
<tr>
<td><strong>Common Core DST 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Introduction 1</strong></td>
<td>Common Core MSc CS - DST (at least 20 EC): Choose 4 out of 9.</td>
<td></td>
</tr>
<tr>
<td>CS4035</td>
<td>Cyber Data Analytics</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Dr.ir. S.E. Verwer</td>
<td></td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/0/2+lab</td>
<td></td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
<td></td>
</tr>
</tbody>
</table>
| **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 five such techniques:  
behavioral profiling and anomaly detection;  
data stream mining and distributed data processing;  
web-crawling and text mining;  
software fuzzing and protocol reverse-engineering; and  
information fusion and collaborative knowledge discovery.  
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 handle the different kinds of input data used to construct these profiles such as websites and software logs. In addition to the traditional sample data sets, software code and implementations form an important source of information for cyber data analytics. In addition to training from execution logs, the student will learn how to use this information source by actively providing input and learning from the returned output. |
| **Study Goals** | The student will be able to:  
Develop and analyze algorithms that learn models from large data streams;  
Detect anomalies in system logs, e.g., for fraud detection;  
Construct behavioral profiles of both people and software;  
Learn insightful models from multiple data sources (e.g., websites, network traces, software code);  
Apply knowledge fusion and collaborative knowledge discovery methods;  
Use machine learning to discover and analyze threats in software components. |
| **Education Method** | There will be two lectures for each of the five topics, and a large lab exercise in which teams of two students will work on a use-case of one of these topics. Each team is free to choose their own topic from a selection of recent research in cyber data analytics. |
| **Assessment** | One large lab assignment in teams of two students resulting in a written report (50%) and an individual summative exam on selected content (50%). |
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.

After three weeks of core topics, the course offers two specialization tracks:
- **MMSR Analytics**, focusing on data analytics aspects for multimedia search and recommendation with special focus on emerging topics.
- **MMSR Systems**, focusing on system and implementation aspects for multimedia search and recommendation with special 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 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;
- explain the difference between topical relevance and utility in multimedia search and recommendation.

In addition to the core goals, students choosing the MMSR Analytics specialization will be able to:
- describe and implement cross-disciplinary approaches to multimedia search and 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.

In addition to the core goals, students choosing the MMSR Systems specialization will be able to:
- describe and implement practical solutions to deal with real-world multimedia search and/or recommendation;
- 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, specialization research or development assignment

### Literature and Study Materials

Will be handed out by lecturers during the course

### Assessment

Written exam over MMSR core and MMSR specialization topics (60%, core and specialization weighted 50/50). A resit exam is offered for this part in Q5.

Specialization assignment for chosen MMSR specialization (week 10/11, 40%):
- For MMSR Analytics: research proposal on an emerging topic in MMSR;
- For MMSR Systems: implementation of a state-of-the-art MMSR research paper.

Depending on the class size, the specialization assignment may be conducted in groups. 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.

The following conditions on admission and grade validity apply:
- A final grade will only be constituted if a 'pass' is obtained for all lab assignments;
- Partial results towards a final course grade (lab, exam, specialization assignment) do not carry over to subsequent academic years.

### Special Information

Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.

### Judgement

Exam on MMSR core and specialization (60%)

Specialization assignment (40%)
### Artificial Intelligence Techniques

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof.dr. C.M. Jonker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0 college</td>
</tr>
<tr>
<td></td>
<td>1/1/0/0 instructie</td>
</tr>
<tr>
<td>Pract.</td>
<td></td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected Prior Knowledge</td>
<td>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), logic (TI1305), and probability theory (TW2215TI). Programming for the first practical assignment will be in Java, for the second in GOAL or Prolog (to be determined).</td>
</tr>
<tr>
<td>Course Contents</td>
<td>Artificial Intelligence techniques for building cognitive agents, and decision support systems are presented. Techniques discussed include automated reasoning, meta-level reasoning, action selection and planning, Markov Decision Processes for optimization, and some aspects of machine learning. In addition, various models needed to design and build such systems are discussed, including cognitive architectures, mental models, decision making, and strategic interaction.</td>
</tr>
</tbody>
</table>
| Study Goals            | After successful completion of the course:  
- Students have a general overview of artificial intelligence  
- Students are able to apply the artificial intelligence techniques discussed  
- Students are able to model knowledge and preferences and using knowledge representation languages.  
- Students are able to design and implement intelligent agents for complex decision making problems. |
| Education Method       | Lectures, tutorials, lab work (practical assignments). |
| Assessment             | Written exam and practical assignments. |
| Remarks                | The lab work will be done in groups of 3 to 4 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. |

### Pattern Recognition

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. D.M.J. Tax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0 + lab 4/4/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected Prior Knowledge</td>
<td>Linear algebra, multivariate statistics.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>After successfully completing this course, the student is able to: recognise pattern recognition 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 pattern recognition problem, using existing software.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures, lab work</td>
</tr>
<tr>
<td>Assessment</td>
<td>Homework, Computer laboratory assignment and written examination. The final grade = 20% homework grade + 40% final computer lab. assignment + 40% written exam.</td>
</tr>
<tr>
<td>Remarks</td>
<td>The exam is an open book examination, except for a multiple choice questions. Only the book can be used, no additional printout of slides, notes, etc. The exam takes 3 hours. You can use a graphical calculated during the exam. Further, no phones, no tables, no laptops or other electronic equipment. For the homework exercises and the final computer lab there is no resit. For the exam there is a resit.</td>
</tr>
</tbody>
</table>

Enrolment / Application Please enrol in Brightspace

Remarks see also http://www.delfileiden.nl/BIO/index.php?id=curriculum

Co-Instructor M. Long
<table>
<thead>
<tr>
<th>IN4086-14</th>
<th>Data Visualization</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Prof. dr. E. Eisemann</td>
<td></td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>A. Vilanova Bartroli</td>
<td></td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/4/0/0 + lab</td>
<td></td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
<td></td>
</tr>
<tr>
<td><strong>Required for</strong></td>
<td>Master course MKE/ST/DS</td>
<td></td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>IN2905-A/IN4152/TI1806 Computer Graphics (recommended, not required).</td>
<td></td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>Basic programming skills are expected, but all relevant topics will be introduced. Basic Java will be used in the second assignment. We consider it requires basic knowledge of programing, but not specifically of java. 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 uses a combination of methods from a very diverse variety of disciplines: perception, computer graphics, human computer interaction, algorithms, image processing, machine learning, numerical analysis, optimization. The course has two main parts information and scientific visualization that require 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>The goal of the course is to get knowledge on the basic fundaments 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, medical, engineering, finances, economics, game analytics.</td>
<td></td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures, practical assignments, self-study, and projects.</td>
<td></td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
<td>Course slides, instructions for projects, and selected literature.</td>
<td></td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>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. 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 couples 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. It is necessary that you register/enroll on Brightspace for this course. 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. 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. 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. 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. The grade consists of 3 elements: Information Visualization project, Volume Visualization Project and a written exam.</td>
<td></td>
</tr>
<tr>
<td><strong>Special Information</strong></td>
<td>In the first lecture, details on the evaluation and practical information on the course will be given.</td>
<td></td>
</tr>
<tr>
<td><strong>Judgement</strong></td>
<td>The project is evaluated based on the developed result, its documentation and presentation. The project is evaluated based on the developed result, its documentation and presentation. The project is evaluated based on the developed result, its documentation and presentation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Final Mark = 0.35 InfoVis Project + 0.35 VolVis Project + 0.3 Exam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The course is passed if the final grade is 6 or higher in average. The exam will get a resit. No resit will be provided for the projects unless the mark on the exam and the other project are above 7.5. It will be evaluated at individual bases, despite the project is done in groups. Resit of a project will mean starting a new project.</td>
<td></td>
</tr>
</tbody>
</table>
Responsible Instructor: Prof. dr. ir. G.J.P.M. Houben

Contact Hours / Week: 2/2/0/0

Education Period: 1

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 course considers science and engineering related to the Web, and in particular Web Data.

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 considers methods and techniques for the design and development of web-based information systems, and as a consequence it gives an insight into the state of the art of the research area of Web Engineering.

As the creation and access of web data often involves properties of the users of that web data, the course also pays attention to research into User Modelling and Adaptive Hypermedia, as the basis for user-adaptation and personalisation in web-based information systems.

In relation to user-adaptation, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling and adaptation. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data.

Other topics that are addressed include the developments concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner, and the developments concerning the role of humans in the processing of web data, e.g. User-Generated Content and Crowdsourcing.

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 research is addressing a whole new range of challenges.

Study Goals: The student learns the important principles and concepts of web-based information systems and their engineering processes, and understands the main research challenges in the area.

The student has knowledge about the main methods, techniques and languages used in the area of web-based information systems, in particular concerning web data.

The student has knowledge of the main principles and techniques for user modelling and adaptation, and of the role of Social Web data 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 40% of the grade, and the large assignment (writing the web science paper), for 60% of the grade. The final paper needs to be graded with a 6.0 or higher. All assignments need to be completed by the indicated deadlines.

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.
<table>
<thead>
<tr>
<th><strong>IN4301</strong></th>
<th><strong>Advanced Algorithms</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Dr. M.M. de Weerdt</td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
<td>Dr. N. Yorke-Smith</td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
<td>Prof. dr. ir. K.I. Aardal</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>2/2/0/0 &amp; lab</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>Basic courses in Algorithmics and Complexity Theory</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>Solving instances of intractable problems in reasonable time. Overview of approximation algorithms and approximation techniques. LP relaxations and semi-definite programming.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>By the end of this course, students will have knowledge of and experience with some advanced algorithmic techniques: exact algorithms for NP-hard problems, approximation techniques, and linear programming relaxation techniques.</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures, homework exercises, and programming assignments.</td>
</tr>
</tbody>
</table>

The expected workload is
- 30% lectures (including preparation for the exams)
- 40% homework exercises
- 30% programming assignments

**Literature and Study Materials**


Supplemental study material will be provided via Blackboard.

**Assessment**

The final mark depends on the marks obtained for:
- (a) homework exercises (6 in total),
- (b) programming assignments (3 in total) and
- (c) the exam (3 parts).

The homework exercises are evaluated on a scale from 0 to 10 and the final mark for the homework exercises (HE) is the average of these results.

Homework exercises have to be completed individually.

There is a programming assignment to be completed at the end of each part. Each assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is a weighted average of the mark obtained for the three assignments with weights of 20%, 40% and 40%, respectively.

Programming exercises can be completed by 2 students working together.

The final exam consists of three parts. Each part will be examined after the four 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.

There is a resit where any subset of the three parts can be redone (advised is a maximum of two). The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. All the (partial) exams and their resits are closed book.

The final mark for the course is determined as follows:
- if each of the HE, PA and EX marks is above 5, the final mark is the average of these three marks.
- if at least one of HE, PA or EX is less than or equal to 5, the final mark is the minimum of the results obtained for HE, PA or EX.

Partial results are valid only in the current academic year.
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.

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.

Interactive lectures, lab assignment, paper presentation and discussion.


No written exams. Lab assignment, paper and presentation.

Twitter handle: https://twitter.com/delftswa

Co-Instructor
M. Finavaro Aniche
### 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.

Topics discussed in class include:

- communication, messaging, naming
- scheduling of distributed workloads, resource provisioning
- fault-tolerance, reliability, availability, fault-tolerant protocols
- consistency, replication, distributed file systems

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.)

### Literature and Study Materials


Additional material: Several relevant research articles introduce the student to the latest advances on the topic.

### Assessment

Overview: The course consists of two main parts: the theoretical Part I and the practical Part II. Students are assessed through a written exam (open questions) and a demonstration of practical skills.

The result of Part I of this course must be at least 6. The result of Part II of this course (practical) must be Completed (C, Voltooid/V in Dutch). The final grade is the result of Part I.
<table>
<thead>
<tr>
<th>Year</th>
<th>2018/2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Electrical Engineering, Mathematics and Computer Science</td>
</tr>
<tr>
<td>Education</td>
<td>Master Computer Science</td>
</tr>
</tbody>
</table>

**Specialisatievakken start kwartaal 1 2018**
<table>
<thead>
<tr>
<th>AP3421</th>
<th>Fundamentals of Quantum Information</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Dr. L. di Carlo</td>
<td></td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. D. Elkouss Coronas</td>
<td></td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>X/X/X/X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4/0/0/0</td>
<td></td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Exam Period</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Knowledge of linear algebra, probability and statistics.</td>
<td></td>
</tr>
<tr>
<td>Course Contents</td>
<td>Approximate syllabus:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- quantum states, unitary operations, and measurements;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- universal gate sets;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- entanglement, Bell test;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- basic quantum communication protocols;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- basic algorithms and quantum algorithmic techniques;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- basic quantum error correction;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- simple physical implementations of qubits.</td>
<td></td>
</tr>
<tr>
<td>Study Goals</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Aim: To learn the fundamental concepts underlying quantum computation and communication systems.</td>
<td></td>
</tr>
<tr>
<td>Education Method</td>
<td>3 hours of lecture, 1 hour tutorial per week.</td>
<td></td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>The main reference textbook for the course is Nielsen and Chuang, Quantum Computation and Information, Cambridge University Press.</td>
<td></td>
</tr>
<tr>
<td>Assessment</td>
<td>40% homework assignments, 10% in class quiz, 50% final exam.</td>
<td></td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>A minimum grade of 5.0 (unrounded) for the final exam is required to pass the course.</td>
<td></td>
</tr>
<tr>
<td>Elective</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Tags</td>
<td>Algorithmics, Physics, Quantum</td>
<td></td>
</tr>
<tr>
<td>Continuing Courses</td>
<td>CS4090 Quantum Communication and Cryptography;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AP3292 Quantum Hardware;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EE4575 Quantum Electronics.</td>
<td></td>
</tr>
</tbody>
</table>
Course Contents
The course introduces algorithms for automated planning and scheduling (P&S) from an Artificial Intelligence perspective.

Possible topics are planning under uncertainty (partially observable Markov decision processes, POMDPs), resource-constrained project scheduling, multiagent planning and temporal planning.

In conjunction with several introductory lectures, students collaborate in small groups on a distinct research project per group, for instance on P&S problems in transport, logistics or smart energy grids. Purely algorithmic challenges will also be provided.

The lectures and research projects provide a good opportunity to learn about topics suitable for Masters projects in the Algorithmics section.

Study Goals
After completing the Algorithms for Planning and Scheduling (P&S) course, the student is able to:
1. Explain general techniques used in P&S algorithms.
2. Explain several specific P&S problem settings and corresponding algorithms.
3. Apply P&S algorithms to problem domains, and can compare and evaluate them.
4. Design and implement an extension of a P&S algorithm.
5. Communicate his/her findings effectively.

Assessment
The assessment consists of the following items:
1. Quality of work of the research project (50%)
2. A scientific report of the research project (including peer review of a report) (20%)
3. Performance during the project (15%)
4. Oral presentation of the research project (15%)

Only items 1 and 2 can be examined a second time.

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.

When possible, students should consider taking courses CS4210-A and CS4210-B instead of this one.

Remarks
The 2018/2019 offering is the final one. This course is being replaced by the following two courses: CS4210-A Algorithms for Intelligent Decision Making and CS4210-B Intelligent Decision Making Project.
**CS4070**
**Multivariate Data Analysis**

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. D.M.J. Tax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Dr. A.J. Cabo</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0 hr, 2/2/0/0 hr pr</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>1</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td></td>
<td>* Basic calculus</td>
</tr>
</tbody>
</table>

**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:
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.

**PART II:**
A course in advanced statistics about linear models in statistics of which the most important ones are: multiple regression analysis and analysis of variance. We treat least squares estimators, testing of linear hypotheses, and the construction of confidence intervals, as well as model validation and diagnostics.

**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 Numbers Central 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.
7. 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.

### 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, exercises and weekly mandatory computer assignments, that are graded.

### Books

**PART I:**

**PART II:**
- John Fox
- Applied Regression Analysis and Generalized Linear Models, 3rd Edition
- 2015 Sage Publications, Inc

### 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. For part (II), the lab counts for 30% and the exam counts for 70%.

### Exam Hours

**PART I:**
- Written exam of 3 hours.

**PART II:**
- Exam consists of the presentation of the analysis of a data set. Depending on the number of students taking the course this will be in a written or oral form (to be decided)

For the time and location of the exams and resits, please consult roosters.tudelft.nl

### Permitted Materials during Tests

**PART I:**
- Self made notes on a two-sided written A4 sheet.
- Calculator.

**PART II:**
- This course is particularly interesting for students that are interested in statistical exploratory and quantitative techniques to analyse multivariate data.
<p>| 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 | The final grade is based on the activity and quality of the participation in the discussions. |
| Permitted Materials during Tests | not applicable |
| Course Contents | The core of social intelligence is our ability to understand and interpret social signals of a person we are communicating with. 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 system that uses social signals in a non-trivial manner, and together evaluate the resulting system. |
| Literature and Study Materials | Selected papers made available before the course. |
| Assessment | Project work (50%): paper, presentation, review of the paper of another group. Theoretical lectures (50%): mini exams, each lecture starts with a min exam on the material discussed in the previous lecture. Project and theoretical work will be averaged. This is your only official final grade. No separate project / theoretical grades will be administered in OSIRIS. No final exam. |</p>
<table>
<thead>
<tr>
<th>CS4200-B</th>
<th>Compiler Construction Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Prof.dr. E. Visser</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>4/4/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Students should take course CS4200-A (Compiler Construction) in parallel with this course.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>Note: This course is the 'project' part of the Compiler Construction course. The 'theory' part (CS4200-A) is required for taking this course. In this course students apply the concepts and techniques studied in CS4200-A to the construction of a compiler and programming environment for a small programming language using the Spoofax Language Workbench.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>After this course students students should be able to:</td>
</tr>
<tr>
<td></td>
<td>- Define the syntax of a programming language using declarative syntax definition that describes the concrete and abstract syntax of a programming language</td>
</tr>
<tr>
<td></td>
<td>- Define transformations on abstract syntax terms to simplify programs</td>
</tr>
<tr>
<td></td>
<td>- Define basic editor services</td>
</tr>
<tr>
<td></td>
<td>- Define the type system (name binding and typing rules) of a programming language using constraint generation rules</td>
</tr>
<tr>
<td></td>
<td>- Define data-flow analyses using control-flow and data-flow rules</td>
</tr>
<tr>
<td></td>
<td>- Define a code generator that translates source language abstract syntax trees to object language instructions using rewrite rules</td>
</tr>
<tr>
<td></td>
<td>- Construct tests for each of the components of a compiler in order to determine its correctness</td>
</tr>
<tr>
<td></td>
<td>- Integrate the components into a working compiler and programming environment</td>
</tr>
<tr>
<td>Education Method</td>
<td>The course runs throughout Semester 1 (September to January) in parallel with CS4200-A in which the theory and basic tutorial assignments for the course are provided.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>The project is conducted using the Spoofax language workbench, which is open source and available from <a href="http://metaborg.org/">http://metaborg.org/</a></td>
</tr>
<tr>
<td>Assessment</td>
<td>Project submissions are doing via pull requests to an individual git repository on a GitLab server.</td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>not applicable</td>
</tr>
<tr>
<td>Judgement</td>
<td>The final grade for the course is the average grade for the milestones of the project.</td>
</tr>
<tr>
<td></td>
<td>We grade the first final solution of each of your assignments. Grades reflect on correctness, clarity, shown programming skills, and readability of your submission. To pass the practical part of the course, you need to meet all of the following criteria:</td>
</tr>
<tr>
<td></td>
<td>- You completed each assignment with a grade of 4.0 or better.</td>
</tr>
<tr>
<td></td>
<td>- You completed the assignments of each milestone with an average grade of 5.0 or better.</td>
</tr>
<tr>
<td></td>
<td>- You completed all the milestones with an average grade of 6.0 or better.</td>
</tr>
<tr>
<td>Course Code</td>
<td>Networking</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>Responsible Instructor</td>
<td>Prof.dr.ir. P.F.A. Van Mieghem</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>4/0/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>2</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
</tbody>
</table>
| 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 |
### Course Contents

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)
  - Definition of ad-hoc networks
  - Comparison with infrastructure based systems
  - Typical applications
  - Advantages and challenges
  - Radio technologies for ad-hoc networks
  - Wi-Fi, Zigbee, Bluetooth
- Modelling ad-hoc networks
  - Propagation models
  - Topology models based on graph theory
  - Degree and hopcount
  - Connectivity theorems
- MAC protocols for ad-hoc networks (Chapter 6, 10 of the textbook)
  - Introduction to MAC protocols
  - Issues and design goals
  - Classification
  - Directional, multi-channel MAC protocols
  - Energy efficiency in MAC protocols
  - Quality of service
- Self organisation and Routing (Chapter 7, 8, 11 of the textbook)
  - Flooding
  - Node discovery, neighbour discovery
  - Route establishment
  - Topology maintenance, localisation
  - Proactive, reactive and hybrid routing
  - Typical protocols
  - Energy efficiency in routing
  - Broadcast and multicast
  - Effects of mobility on connectivity and capacity
  - Effect of nodes joining and leaving the network
- Advanced issues in ad hoc networks
  - Wireless sensor networks (Chapter 12 of the textbook and papers)
  - Cooperation (Reference papers)
  - Simulating ad hoc networks as part of project (optional: ns3, OMNET, OPNET)
  - Energy Harvesting

### Project presentations by students

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

### Literature and Study Materials

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)
   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 the 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/Blackboard/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 2%

Permitted Materials during Tests
Different conditions for different tests.
Conditions will be informed 1 week before the exams/test.

IN4010(-12) Artificial Intelligence Techniques

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof. dr. C.M. Jonker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Dr. F.A. Oliehoek</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0 college</td>
</tr>
<tr>
<td></td>
<td>1/1/0/0 instructie</td>
</tr>
<tr>
<td>Pract.</td>
<td></td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>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), logic (TI1305), and probability theory (TW2215TI). Programming for the first practical assignment will be in Java, for the second in GOAL or Prolog (to be determined).</td>
</tr>
<tr>
<td>Course Contents</td>
<td>Artificial Intelligence techniques for building cognitive agents, and decision support systems are presented. Techniques discussed include automated reasoning, meta-level reasoning, action selection and planning, Markov Decision Processes for optimization, and some aspects of machine learning. In addition, various models needed to design and build such systems are discussed, including cognitive architectures, mental models, decision making, and strategic interaction.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>After successful completion of the course: - Students have a general overview of artificial intelligence - Students are able to apply the artificial intelligence techniques discussed - Students are able to model knowledge and preferences and using knowledge representation languages. - Students are able to design and implement intelligent agents for complex decision making problems.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures, tutorials, lab work (practical assignments).</td>
</tr>
<tr>
<td>Assessment</td>
<td>Written exam and practical assignments.</td>
</tr>
<tr>
<td>Remarks</td>
<td>The lab work will be done in groups of 3 to 4 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.</td>
</tr>
</tbody>
</table>
### IN4085  Pattern Recognition

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. D.M.J. Tax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor</td>
<td>Dr. D.M.J. Tax</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0 + practical</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>2</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>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 a preferred but not required.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>After successfully completing this course, the student is able to: recognise pattern recognition 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 pattern recognition problem, using existing software.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures, lab work</td>
</tr>
<tr>
<td>Computer Use</td>
<td>Lab exercises: implementing (small) parallel programs with C and MPI.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Will be made available throughout the course and can be downloaded from the Blackboard.</td>
</tr>
<tr>
<td>Assessment</td>
<td>Written exam (50%) + Lab work (50%)</td>
</tr>
<tr>
<td>Enrolment / Application</td>
<td>Via Osiris</td>
</tr>
</tbody>
</table>

---

### IN4097TU  Introduction to High Performance Computing

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof. dr.ir. H.X. Lin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0 and practical</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>2</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>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 a preferred but not required.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>High Performance Computing, Parallel programming, parallel computing</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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; 5. Obtain some experience on performance analysis of parallel programs.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures, computer lab exercise using MPI</td>
</tr>
<tr>
<td>Computer Use</td>
<td>Lab exercises: implementing (small) parallel programs with C and MPI.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Will be made available throughout the course and can be downloaded from the Blackboard.</td>
</tr>
<tr>
<td>Assessment</td>
<td>Written exam (50%) + Lab work (50%)</td>
</tr>
<tr>
<td>Enrolment / Application</td>
<td>Via Osiris</td>
</tr>
</tbody>
</table>

---

### Course Details

**Responsible Instructor:** Z. Erkin  
**Contact Hours / Week:** x/x/x/x  
**Education Period:** 1  
**Start Education:** 1  
**Exam Period:** 2  
**Course Language:** English  
**Required for:** UT-201500042 Privacy Enhancing Technologies (Q4)  
**Expected prior knowledge:** Basic understanding on the following is needed.  
- Probability and statistics  
- Discrete Mathematics or Modular Arithmetic  
- 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 practical experience on the topics covered.

### Education Method

Lectures, assignments and 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 45 minutes sessions, total 22 hours
- Practice session: 7 x 90 min. total 12 hours
- Assignments: 4 x 15 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)

**Assessment:**
- Handouts of lectures
- written exam(60%) + mandatory assignments (40%)

**Exam Hours**
The exam is closed book.

**Permitted Materials during Tests**
Only non-scientific calculators.
<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof.dr.ir. G.J.P.M. Houben</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>2/2/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>The expected entry level is (equivalent to) standard bachelor-level computer science.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>The course considers science and engineering related to the Web, and in particular Web Data. 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 considers methods and techniques for the design and development of web-based information systems, and as a consequence it gives an insight into the state of the art of the research area of Web Engineering. As the creation and access of web data often involves properties of the users of that web data, the course also pays attention to research into User Modelling and Adaptive Hypermedia, as the basis for user-adaptation and personalisation in web-based information systems. In relation to user-adaptation, the Social Web plays a major role, for example because data from the social web creates a great source of knowledge for user modelling and adaptation. Therefore, the course also considers research in social web data analytics and data science techniques to extract user knowledge from social web data. Other topics that are addressed include the developments concerning the Semantic Web, with its main languages, theory and applications and tools for describing semantics in machine-processable manner, and the developments concerning the role of humans in the processing of web data, e.g. User-Generated Content and Crowdsourcing. 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 research is addressing a whole new range of challenges.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>The student learns the important principles and concepts of web-based information systems and their engineering processes, and understands the main research challenges in the area. The student has knowledge about the main methods, techniques and languages used in the area of web-based information systems, in particular concerning web data.</td>
</tr>
<tr>
<td>Education Method</td>
<td>The education includes:</td>
</tr>
<tr>
<td></td>
<td>- Lectures, before which and after which students study material by themselves, to get an understanding of the relevant material;</td>
</tr>
<tr>
<td></td>
<td>- Small assignments and hands-on exercises, to apply the understanding of relevant material;</td>
</tr>
<tr>
<td></td>
<td>- One large assignment, with a number of feedback moments, to learn how to write a web science paper and contribute to relevant research.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Will be provided in class, depending on the topics chosen for the assignments and final paper.</td>
</tr>
<tr>
<td>Assessment</td>
<td>Assessment happens on the basis of the small assignments (accompanying the lectures), for 40% of the grade, and the large assignment (writing the web science paper), for 60% of the grade. The final paper needs to be graded with a 6.0 or higher. All assignments need to be completed by the indicated deadlines.</td>
</tr>
<tr>
<td>Special Information</td>
<td>Students are asked to register/enrol on Brightspace.</td>
</tr>
<tr>
<td>Remarks</td>
<td>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.</td>
</tr>
<tr>
<td><strong>IN4301</strong></td>
<td><strong>Advanced Algorithms</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Dr. M.M. de Weerdt</td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
<td>Dr. N. Yorke-Smith</td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
<td>Prof. dr. ir. K.I. Aardal</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>2/2/0/0 &amp; lab</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>Basic courses in Algorithmics and Complexity Theory</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>Solving instances of intractable problems in reasonable time. Overview of approximation algorithms and approximation techniques. LP relaxations and semi-definite programming.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>By the end of this course, students will have knowledge of and experience with some advanced algorithmic techniques: exact algorithms for NP-hard problems, approximation techniques, and linear programming relaxation techniques.</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures, homework exercises, and programming assignments.</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>The final mark depends on the marks obtained for (a) homework exercises (6 in total), (b) programming assignments (3 in total) and (c) the exam (3 parts). The homework exercises are evaluated on a scale from 0 to 10 and the final mark for the homework exercises (HE) is the average of these results. Homework exercises have to be completed individually. There is a programming assignment to be completed at the end of each part. Each assignment is graded on a scale from 0 to 10. The final mark for the programming assignment (PA) is a weighted average of the mark obtained for the three assignments with weights of 20%, 40% and 40%, respectively. Programming exercises can be completed by 2 students working together. The final exam consists of three parts. Each part will be examined after the four 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. There is a resit where any subset of the three parts can be redone (advised is a maximum of two). The result for a part after the resits is determined by the maximum score obtained for the original exam and the resit. All the (partial) exams and their resits are closed book. The final mark for the course is determined as follows: - if each of the HE, PA and EX marks is above 5, the final mark is the average of these three marks. - if at least one of HE, PA or EX is less than or equal to 5, the final mark is the minimum of the results obtained for HE, PA or EX. Partial results are valid only in the current academic year.</td>
</tr>
<tr>
<td>IN4307</td>
<td>Medical Visualization</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>A. Vilanova Bartroli</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>2/0/0 lectures &amp; 4/0/0 lab.</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>Theory and practice of medical visualization. This includes the following aspects: data acquisition basics, clinical practice; image processing: filtering, segmentation and measurement, registration; medical volume visualization; illustrative visualization; advanced visualization for complex modalities; interaction techniques for medical data; advanced applications.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>At the end of the course, the students should be able to understand, and judge the advantages and disadvantages of the medical visualization algorithms, as well as their applicability to a specific medical problem. The students should be able to propose suitable solutions to a problem, backed by sound knowledge of the underlying theory and the practical possibilities. They should be able to design, implement, test and discuss these solutions, consisting of a number of medical visualization algorithms.</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>The course will be based on a combination of lectures and practical assignments.</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>The evaluation will be based on a final project (60%) and an written (or oral if few number of students allow) exam (40%). The final project will be done during the 2nd quarter. The deliverables for the final project will be a report (paper), the results (e.g., code) and a presentation. There will be a day in the second quarter were all projects will be presented. 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 written/oral exam. No computer or laptop is allowed. Both Assignment (60%) and written/oral exam (40%) have to get the mark of pass to successfully pass the course. Exam has a resit. Project will have a resit if the exam has been passed with a mark of 7 or higher.</td>
</tr>
<tr>
<td><strong>Permitted Materials during Tests</strong></td>
<td>Notes and written material. No computers.</td>
</tr>
<tr>
<td><strong>Special Information</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
IN4309  Random Signal Processing

<table>
<thead>
<tr>
<th>Study Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Probability Theory</td>
</tr>
<tr>
<td>- Conditional probabilities, the law of total probability, and Bayes rule.</td>
</tr>
<tr>
<td>- Solve probability problems that require the use of axioms of probability.</td>
</tr>
<tr>
<td>2. Definition and Description of Random Variables and Processes</td>
</tr>
<tr>
<td>- 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.</td>
</tr>
<tr>
<td>- Convert the description of a probabilistic problem into a probabilistic model using PDF, PMF, or CDF.</td>
</tr>
<tr>
<td>3. PDF/PMF and Expected Value</td>
</tr>
<tr>
<td>- Calculate the various forms of expected value of (combinations of) random variables and random processes</td>
</tr>
<tr>
<td>- 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.</td>
</tr>
<tr>
<td>- Calculate the PDF, PMF, expected value and variance of a derived random variable.</td>
</tr>
<tr>
<td>4. Properties of Random Processes</td>
</tr>
<tr>
<td>- Independence, orthogonality, uncorrelated, whiteness, IID - Determine if random variables/processes have the following properties: independent, orthogonal, uncorrelated, white, Poisson, Gaussian, Bernoulli, IID, stationary, WSS, ergodic.</td>
</tr>
<tr>
<td>- 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.</td>
</tr>
<tr>
<td>5. Large Numbers/Central limit theorem, law of large numbers</td>
</tr>
<tr>
<td>- Solve problems that require the use of the central limit theorem in an engineering context</td>
</tr>
<tr>
<td>- Explain the law of the large numbers in an engineering context.</td>
</tr>
<tr>
<td>6. Statistical Estimators</td>
</tr>
<tr>
<td>- Estimated mean, variance, and correlation function</td>
</tr>
<tr>
<td>- Given a set of outcomes, sample functions or realizations, calculate estimators for expected value, variance, and (auto-)correlation function.</td>
</tr>
<tr>
<td>7. Application to Engineering Problems and Simulations</td>
</tr>
<tr>
<td>- Select and translate a simple electrical engineering or computer science problem into mathematical probability model.</td>
</tr>
<tr>
<td>- 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.</td>
</tr>
<tr>
<td>- 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.</td>
</tr>
<tr>
<td>8. Signals and Systems</td>
</tr>
</tbody>
</table>

**Course Contents**

Signal processing plays a major role in many applications, like consumer electronics (mp3 player, mobile telephony, CD player, (HD)TV), radar and medical applications.

In this course we will discuss fundamental signal processing principles, methods and algorithms. The course consists of two parts: Part I: Stochastic Processes (given in Q1) and Part II: Digital Signal Processing (given in Q2).

The first part of the course introduces the concept of stochastic models and random processes, while the second part of the course focusses on deterministic signal processing. The stochastic models are used for describing systems and signals that are not deterministic. In fact, no single system or signal is deterministic in practice. For that reason the theory of stochastics and random processes should be considered as a useful extension of known approaches for describing and modeling systems and signals encountered in engineering practices.

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. The part on deterministic signal processing will cover the representations of signals and systems in time-, frequency (Fourier) and Z-domains and the foundations of mathematical models for signal analysis and processing based on these representations.

The course addresses the following subjects:

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.
7. Refresher (discrete-time) signals and systems.
10. Sampling and reconstruction of signals.

**Expected Prior Knowledge**

This course builds heavily on Probability and Statistics, Mathematics (especially integration and differentiation), Signal Processing and Transformations (especially linear systems and signals, Fourier analysis).

**Course Language**

English

**Contact Hours / Week**

1/1/1/0/0/0

**Start Education**

1

**Exam Period**

1/2/2/3/1/0

**Responsible Instructor**

Prof. Dr. A. Hanjalic

Dr. D.M.J. Tax
9. Z-transform
- Properties of Z-transform, region of convergence, rational transfer functions, inverse Z-transform, system analysis in the Z-domain, poles, zeros, stability.

10. Fourier Transforms
- Fourier series, continuous-time Fourier transform, discrete-time Fourier transform, discrete Fourier transform, Fast Fourier transform (FFT), properties of Fourier transforms, frequency-domain characterization of LTI systems.

11. Sampling and Reconstruction of Signals
- Sampling theorem, sampling frequency, aliasing, folding, interpolation, D/A and A/D conversion.

**Education Method**
- Lectures, working groups (problem solving), laboratory work (Matlab exercises)

For Part I: 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).

**Literature and Study Materials**


**Assessment**
The two parts will be examined separately, and both need to be passed.
Exams for both parts are closed book, but students are allowed to bring a self made double A4 formula sheet.

Before students can take part in the exam for part I, the students have to hand in pass the final laboratory exercise.

---

<table>
<thead>
<tr>
<th>IN4310 Seminar Computer Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
</tr>
</tbody>
</table>
### Seminar Selected Topics in Multimedia Computing

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>P.S. Cesar Garcia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>2/0/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>signal (image, audio) processing, pattern recognition, networking and distributed systems</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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).</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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 &quot;hot topic&quot;, such that they are able to identify the &quot;knowledge gap&quot; (i.e., the place in which more research is needed in order to advance the state of the art).</td>
</tr>
<tr>
<td>Education Method</td>
<td>readings, seminar discussions, presentations, survey paper</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Readings, possibly including video lectures.</td>
</tr>
<tr>
<td>Assessment</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Seminar Web Information Systems

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof.dr.ir. G.J.P.M. Houben</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>2/0/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>1</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Standard bachelor-level computer science or equivalent.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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 services, 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 content mining, web information filtering and retrieval), -social web (e.g. social web data analytics, Web 2.0, 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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>-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.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Student seminar.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Is provided in the seminar, depending on the chosen subjects.</td>
</tr>
<tr>
<td>Assessment</td>
<td>-Quality of presentation of the scientific paper studied (15%). -Participation in the seminar discussions (10%). -Quality of paper written (75%).</td>
</tr>
<tr>
<td>Special Information</td>
<td>Students are asked to register/enrol on Blackboard. Students are also asked to be present and active in the first seminar session, to facilitate the proper planning of the seminar.</td>
</tr>
<tr>
<td>Remarks</td>
<td>The expected workload of Sects 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.</td>
</tr>
<tr>
<td>Maximum number of participants</td>
<td>This course has a maximum capacity of 50 students. Students of 1)Web Information Systems group and 2) EEMCS have priority for other students.</td>
</tr>
</tbody>
</table>
### Analytics and Machine Learning for Software Engineering

<table>
<thead>
<tr>
<th>负责教师</th>
<th>Dr. ir. G. Gousios</th>
</tr>
</thead>
<tbody>
<tr>
<td>学时/周</td>
<td>5/0/0/0</td>
</tr>
<tr>
<td>教育周期</td>
<td>1</td>
</tr>
<tr>
<td>考试周期</td>
<td>none</td>
</tr>
<tr>
<td>课程语言</td>
<td>English</td>
</tr>
</tbody>
</table>

#### 预期先修知识
- 熟悉编程是必要的。
- 熟悉研究方法是可选的。

#### 课程内容
软件仓库是软件工程数据的宝库，包括源代码、执行踪迹、历史代码变更、邮件列表和bug报告。这些数据包含关于项目状态和历史的丰富信息。对软件仓库进行数据科学分析，研究人员可以了解软件开发实践的实证基础，并使实践者能更好地管理、维护和演变复杂的软件项目。

#### 学习目标
本课程旨在探讨软件工程数据挖掘中的技术和发展前沿，讨论数据挖掘软件工程数据的挑战，突出成功案例，并概述未来的研究方向。学生将获得在研究或实践中进行数据科学分析所需的知识。

#### 教育方法
课程是研讨会，这意味着我们将研究软件工程分析领域的文献。本课程包括以下教育方法：
- 专家讲座
- 自主学习和研究论文的自我展示
- 开发工具用于从软件仓库提取数据

#### 学术和学习材料
讲座材料和研究论文是课程的基础。教师将共享这些资源作为学习材料。

#### 参考文献

#### 评估
最终成绩由以下内容组成：
- 50% - 调查贡献的质量
- 30% - 实现CodeFeedr插件
- 20% - 相关工作的展示

课程没有考试，也不允许重考任何内容。

#### 入学/申请
每个想参加本课程的学生成为"要求"的：
- 在Brightspace上注册/参加课程前参加第一堂课

#### 标签
- Databases
- Programming
- Project
- Research Methods
- Software
- Software Engineering
### IN4387: System Validation

<table>
<thead>
<tr>
<th><strong>Responsible Instructor</strong></th>
<th>Dr. ir. J.J.A. Keiren</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>2/0/0/0 HC 4/0/0/0 instr &amp; lab</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Required for</strong></td>
<td>Embedded Systems Masters</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Parts</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>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 notoriously hard to detect such bugs using testing (timing, e.g., 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</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>Behavioural Specification using Process Theory (Labelled Transition Systems, various notions of behavioural equivalence) and process algebra. Model checking the modal mu-calculus, and model-based testing using IOCO.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>Upon completion of the course: 1. The student knows the fundamental theory necessary for specifying the behaviour of embedded systems and for reasoning about this behaviour. 2. The student can describe simple systems using this theory. 3. The student can formally specify requirements and prove (or disprove) them on the behaviour. 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.</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures + Practical Project</td>
</tr>
<tr>
<td><strong>Computer Use</strong></td>
<td>The lectures are held in the first quarter after which a written exam (on the theory treated in the lectures) is taken. Parallel to the theory part, a practical project is done. The project is carried out in groups of about 4 students and the result is a verified model of an embedded system together with a comprehensive report on the steps towards to the model.</td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
<td>The course is based on the book by Groote and Mousavi (see &quot;Books&quot;). All other materials will be published on Brightspace.</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>The result of this course will be based upon the results of the written examination (50%) and the practical project (50%). For both the written examination and the practical project a minimum of 5.0 is required in order to pass 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.</td>
</tr>
<tr>
<td><strong>Permitted Materials during Tests</strong></td>
<td>At the exam an appendix with the axioms for process algebra will be provided. Other material may not be used.</td>
</tr>
<tr>
<td><strong>Enrolment / Application</strong></td>
<td>Brightspace</td>
</tr>
<tr>
<td><strong>Remarks</strong></td>
<td>The study load of the course is 5 ECTS, which is 5 x 28 = 140 hours. Of these, about 20 are spent in lectures, roughly 70 hours will be spent on the project; note that this amounts to 10 hours per week. There are 50 hours left for self-study activities. Given that the course takes 7 weeks, this means students are expected to spend between 6 and 7 hours per week on average outside the lectures and project work on studying the material and preparing for the exam.</td>
</tr>
</tbody>
</table>
Cloud computing has emerged in the past few years as an IT paradigm in which the infrastructure, the platform, and even the software used by IT operations are outsourced services. Departing from the IT services of the past two decades, cloud services can flexibly be used when and only for how long they are needed, and leased for prices charged in small increments of actual usage. The cloud computing model impacts favorably the operation of small and medium sized enterprises (SMEs), by reducing the initial investment in IT and the maintenance overheads, and by allowing SMEs to focus on developing their core technology and business, rather than becoming expert IT operators. There are currently hundreds of commercial cloud-based service providers, such as Amazon, Microsoft, Google, and SalesForce. This course focuses on the cloud computing paradigm.

Course Topics
1. Overview of cloud computing concepts, objectives, and architectures (2h).
2. Scheduling and resource management in clouds (2h).
3. Data center and energy efficiency (2h).
4. Multi-tenancy concepts, including virtualization (2h).
5. Cloud programming models (2h).
6. Case studies (2h).
7. Open research challenges with a guest lecturer (2h).

Note: the course topics are accompanied by 1-3 recent research short articles for students to read and review.

Study Goals
1. Explain the basic concepts, objectives, and functions of cloud computing.
2. Describe the architecture and operation of cloud computing.
3. Describe the elements of user workloads in cloud computing.
4. Explain how cloud computing can schedule the workloads of multiple users (virtualization, multi-tenancy).
5. Describe the programming models applicable for cloud computing.
6. Implement complex operations of cloud computing in realistic scenarios.
7. Analyze the tradeoffs inherent in the design of cloud computing data centers and applications.

Education Method
Language: English
Lectures and reviews: in-class, 7 weeks x 2h + self-study, 40h: at least 6 articles of the offered 12-15
Seminar: self-study, 28h + 20 minutes: Presentation on selected topic (once)
Practical: Groups of 2 on the DAS distributed computer system.
three exercises in-class, 6 weeks x 2h + one large exercise of 40h: large exercise based on course topics 6 and 7 + report of 46 pages

Assessment
Papers to be read for the paper reviews
Overall: Assessment through: in-class presentation, portfolio of reviews of self-study material, and demonstration of practical ability through three minor and one major assignment.
Note: There is no final exam.

Summary:
Type of assessment Part of final grade
Presentation 20% Reviews 30% Practical 50%
 Responsible Instructor | R.R. Venkatesha Prasad
---|---
Contact Hours / Week | 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 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.
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) lecture presentation on a selected research paper (from top journals/conferences) presented individually by students and, (ii) work on an associated 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 needs to be used and demonstrated. User experience where applicable also needs to be done.
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 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. 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. Part 2: a report describing the outcome of the assigned project. Part 1: 30% of the whole mark; Part 2: 60% of the whole mark. Points would In the assessment, a focus on the practicality and entrepreneurial aspect of the idea will be prevailing.
10% marks would be awarded if the report and work is ready for a publication.
Please NOTE: There would be no resist since this is a hands on work based course.
Tags | Circuits
Group work
Programming
Project
Software
maximum aantal deelnemers | Max 30 students, which translates to max of 10 groups (in case of odd number one group composed of 3 students).

 Responsible Instructor | Dr.ir. J.F.M. Tonino
Contact Hours / Week | 0/0/6/0
Education Method | Capita Selecta is a custom course designed by a scientific staff member to provide students an opportunity for more in-depth study of selected material.
When submitting the Individual Study Program to the Board of Examiners, the Capita Selecta description needs to be attached, including how it will be assessed. The course might be taken as a Specialisation course or a Free Elective course.
| Specialistievakken start kwartaal 2 2018 |
### CS4015  Behaviour Change Support Systems

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. ir. W.P. Brinkman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/4/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>2</td>
</tr>
<tr>
<td>Start Education</td>
<td>2</td>
</tr>
<tr>
<td>Exam Period</td>
<td>2</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>none</td>
</tr>
<tr>
<td>Course Contents</td>
<td>Behavior change support systems (BCCS) 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 personalized financial guidance are three examples of these systems. To establish, modify or maintain change BCCS 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. BCCS are found in many domains, including education, sales, negotiation, management, and particular in the health domain.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>The course allows students to achieve understanding of principles, concepts and theories underlying BCCS systems and methods for designing them.</td>
</tr>
<tr>
<td>Education Method</td>
<td>In the lectures, theories, principles and methods are presented and discussed. Students will work in small groups on their own design for a BCCS. The students will present these ideas during the lectures.</td>
</tr>
<tr>
<td>Expected Workload</td>
<td>Lectures: 22 hours (11 × 2 hours lectures)</td>
</tr>
<tr>
<td></td>
<td>Reading time: 44 hours (11 × 4 hours reading time and watching videos for each lecture)</td>
</tr>
<tr>
<td></td>
<td>Preparation presentation: 15 hours (3 × 5 hours for each presentation)</td>
</tr>
<tr>
<td></td>
<td>Coursework project, including writing report: 40 hours</td>
</tr>
<tr>
<td></td>
<td>Exam preparation and revision: 19 hours</td>
</tr>
<tr>
<td></td>
<td>Total = 140 hours</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Will be announced on blackboard</td>
</tr>
<tr>
<td>Assessment</td>
<td>The module is assessed by coursework and an written exam as follows: (60%) Written Exam (40%) Coursework Project (resulting in a report handed in at the end of Q2) This will be combined into a single mark. To pass the module, students need to pass both the written exam and coursework part.</td>
</tr>
<tr>
<td>Exam Hours</td>
<td>3</td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>none</td>
</tr>
</tbody>
</table>

### CS4055  High Performance Data Networking

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. ir. F.A. Kuipers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/4/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>2</td>
</tr>
<tr>
<td>Start Education</td>
<td>2</td>
</tr>
<tr>
<td>Exam Period</td>
<td>2</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Basic understanding of networking and programming.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>The high-performance data networking course is an advanced networking course that treats concepts like network design, Quality of Service, network robustness, and addresses state-of-the-art technologies, such as optical networks, software-defined networking, and information-centric networking.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>The learning objectives of this course are twofold: (1) The student should have a passive to active knowledge of the treated networking technologies. (2) The student should be able to apply and work with the technologies in a network lab/emulator/simulator.</td>
</tr>
<tr>
<td>Education Method</td>
<td>The first half of the course will consist of lectures (4x4 hours), self-study (38 hours) and an exam (2 hours), and the second half focuses on instruction (4x4 hours) and individual project work (68 hours).</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Slides</td>
</tr>
<tr>
<td>Assessment</td>
<td>The final assessment is based on (1) the project result(s) and (2) an exam, each counting for 50% of the final grade. No resit is scheduled. Instead, additional project work can be done to pass the course.</td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>No books, slides, notes, etc. are permitted for the exam. Only a pen is needed/ permitted.</td>
</tr>
<tr>
<td>CS4090</td>
<td><strong>Quantum Communication and Cryptography</strong></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Prof. dr. S.D.C. Wehner</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>weekly: 3 hours lecture, 1 hour tutorial</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>Linear Algebra, Probability &amp; Statistics, Q101 (Fundamentals of quantum information)</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>This class will teach you all about quantum cryptography! This class is given in an inverted classroom style. It will consist of an online component on edX QuCryptoX - as well as a contact session every week where we will discuss examples and exercises. See <a href="https://www.edx.org/course/quantum-cryptography-caltechx-delftx-qucryptox">https://www.edx.org/course/quantum-cryptography-caltechx-delftx-qucryptox</a> for course contents. Caution: This class requires you to take “Fundamentals of Quantum Information” in Quarter 1.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>The student will acquire: A good understanding of the fundamental concepts of quantum information theory A good understanding of the essential tools in quantum cryptography 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)</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures and Tutorials</td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
<td><strong>Primary:</strong> Lecture Notes <strong>Auxilliary:</strong> Nielsen and Chuang Quantum computation and information, Cambridge University Press. Mark Wilde Quantum information theory, Cambridge University Press</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>Final Exam and Pass at EdX Exercises (Weekly homework, quick online quiz and short literature study)</td>
</tr>
<tr>
<td><strong>Tags</strong></td>
<td>Abstract Adventurous Algorithmics Challenging Group Dynamics/Project Organisation Information &amp; Communication Integrated Intensive Involved Lineair Algebra Mathematics Physics Quantum Signals Technology Telecommunication Transport phenomena</td>
</tr>
</tbody>
</table>
**Course Contents**

**Motivation:**
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-formed code). Programming languages can help developers to prevent programming errors like these by defining coding principles and detecting violations of those principles through dynamic and static code analysis. Such language-based countermeasures relieve software developers of part of the burden of ensuring software security. But how to select and apply language-based countermeasures?

**Synopsis:**
This course studies dynamic and static code analysis techniques as language-based countermeasures to security vulnerabilities. In particular, we will investigate and compare the trade-offs of the following countermeasures:

- **Dynamic analysis:** Run-time monitoring
- **Dynamic analysis:** Compile-time instrumentation
- **Static analysis:** Type systems
- **Static analysis:** Data-flow analysis
- **Static analysis:** Abstract interpretation

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**
Students are able to:

- Describe the nature of security vulnerabilities in software systems.
- Explain different language-based countermeasures to security vulnerabilities and compare their respective trade-offs.
- Formally define variations of the dynamic and static analyses discussed in class.
- Contrast programming languages based on the set of countermeasures they provide.

**Education Method**
Lectures + lab assignments + reading assignments

**Assessment**
Homework assignments and oral or written exam

**Tags**
Programming
Programming concepts
Programming Software
Software
Software Engineering
<table>
<thead>
<tr>
<th>CS4160 Blockchain Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
</tr>
<tr>
<td><strong>Instructor</strong></td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
</tr>
<tr>
<td><strong>Co-Instructor</strong></td>
</tr>
</tbody>
</table>
Data Visualization

Responsible Instructor
Prof.dr. E. Eisemann
A. Vilanova Bartroli

Contact Hours / Week
0/4/0/0 + 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/TI1806 Computer Graphics (recommended, not required).
Basic programming skills are expected, but all relevant topics will be introduced. Basic Java will be used in the second assignment. We consider it requires basic knowledge of programming, but not specifically of java.

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 uses 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 require 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 basic fundaments 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, medical, engineering, finances, economics, game analytics.

After the course, the student has knowledge and understanding of a wide range of general visualization techniques, perception principles, mathematical foundations, algorithms, and relevant data representations. Furthermore, the student is able to adapt, apply and develop suitable techniques for a given practical visualization 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

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 couples 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.

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

The course is passed if the final grade is 6 or higher in average.

The exam will get a resit. No resit will be provided for the projects unless the mark on the exam and the other project are above 7.5. It will be evaluated at individual bases, despite the project is done in groups. Resit of a project will mean starting a new project.
**IN4150**  
**Distributed Algorithms**  

<table>
<thead>
<tr>
<th><strong>Field</strong></th>
<th><strong>Details</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Prof. dr. ir. D.H.J. Epema</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/4/0/0 lecture; lab 2nd period.</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
</tbody>
</table>
| **Expected prior knowledge** | Operating system concepts (TI2726C)  
Computer Networks (IN2605) |
| **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**         | After having completed this course, the student has a good knowledge of and insight into important fundamental (theoretical) problems in distributed systems and their algorithmic solutions. In addition, the student can design and implement distributed algorithms that solve these problems. |
| **Education Method**    | Lectures, lab work                                                          |
| **Literature and Study Materials** | Lecture notes (on Brightspace) |
| **Assessment**          | One paper review and a written exam (closed book). The grade of the exam is the grade for the course. |
| **Remarks**             | Lab work is 40 hrs.                                                         |

**IN4302TU**  
**Building Serious Games**  

<table>
<thead>
<tr>
<th><strong>Field</strong></th>
<th><strong>Details</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Dr. ir. A.R. Bidarra</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/4/0/0</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>Different, to be announced</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
</tbody>
</table>
| **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, 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. 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 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  
o deepening programming skills while building a complex and large software system in an agile context |
| **Education Method**    | Project  
Also a few plenary sessions and/or lectures |
| **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 submitting it to a conference on serious games and/or their application. |
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Performance Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Prof.dr.ir. P.F.A. Van Mieghem</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/2/0/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>2</td>
</tr>
<tr>
<td>Start Education</td>
<td>2</td>
</tr>
<tr>
<td>Exam Period</td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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 &quot;Quality of Service (QoS)&quot; 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 blackboard.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures and homework after each class</td>
</tr>
<tr>
<td>Assessment</td>
<td>Written and closed book. A formularium is provided that can be consulted at the examination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Year</th>
<th>2018/2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Electrical Engineering, Mathematics and Computer Science</td>
</tr>
<tr>
<td>Education</td>
<td>Master Computer Science</td>
</tr>
</tbody>
</table>

**Specialistievakken start kwartaal 3 2018**
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. Morphological 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, mean shift), model-driven image segmentation (Hough transform, template matching, deformable templates, active contours, ASM/AAM, level sets).

Study Goals
General learning goals of the course are:

1. acquiring in-depth knowledge of state-of-the-art image processing techniques;
2. being able to solve elementary problems related to the theory mentioned in 1;
3. being able to solve more advanced problems addressing the theory mentioned in 1 by combining mathematical skills and physical insight;
4. being able to acquire new knowledge about an image processing technique;
5. being able to present newly acquired knowledge about a medical imaging technique.

Education Method
Lectures, group assignment with plenary presentation and discussion.

Computer Use
Matlab including the dipimage toolbox and/or other image processing toolboxes.

Literature and Study Materials


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/c3gfn/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. The final grade is the average of the two parts.
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
<table>
<thead>
<tr>
<th>Course Contents</th>
<th>Software is one of the most complex artifacts of mankind has ever created, but complexity is the enemy of correctness. Modern software testing tools use a multitude of automated techniques geared toward correct computer code, amongst others:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Concrete and symbolic (concolic) testing</td>
</tr>
<tr>
<td></td>
<td>Execution monitoring and taint analysis</td>
</tr>
<tr>
<td></td>
<td>Mutation testing</td>
</tr>
<tr>
<td></td>
<td>Reverse-engineering and binary analysis</td>
</tr>
<tr>
<td></td>
<td>Search-based and model-based test case generation</td>
</tr>
<tr>
<td></td>
<td>State machine learning or inference</td>
</tr>
<tr>
<td></td>
<td>Web testing</td>
</tr>
<tr>
<td>Study Goals</td>
<td>The student will acquire:</td>
</tr>
<tr>
<td></td>
<td>Understanding of different advanced software testing techniques</td>
</tr>
<tr>
<td></td>
<td>Ability to test and improve the correctness of existing systems</td>
</tr>
<tr>
<td></td>
<td>Ability to analyze and reverse-engineer software code</td>
</tr>
<tr>
<td>Education Method</td>
<td>The main part of the course will consist of a lab assignment where the students will need to test a real software system using a state-of-the-art techniques described in the scientific literature. This will be supported by a few lectures covering the tools and techniques that are required for these tasks, such as SAGE, Pitest, CWSandbox, LearnLab, Torx, Sulley, Crawljax, Valgrind, and Z3. There will be many instruction sessions where students can work on their assignment and ask the teachers for assistance.</td>
</tr>
<tr>
<td>Assessment</td>
<td>Lab work 50% including a written report 30% and presentation of the results 20%.</td>
</tr>
<tr>
<td>Course Title</td>
<td>Seminar Research Methodology for Data Science</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Responsible Instructor</td>
<td>Dr. ir. W.P. Brinkman</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. K.A. Hildebrandt</td>
</tr>
<tr>
<td>Instructor</td>
<td>J. Urbano Merino</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/X/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Start Education</td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>basic knowledge in mathematics (linear algebra, calculus, probability and statistics)</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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 use of tools to do this. The main topics of study considered are: Conceptualizing research questions and experimental design Generalized linear models for statistical analysis Multilevel modeling for hierarchical and longitudinal data analysis Linear problem modeling in data processing Non-linear problem modeling in data processing Numerical solvers and applications Measuring and sampling, validity and reliability</td>
</tr>
<tr>
<td>In the course, students will be using software tools such as R, and Matlab/Mathematica</td>
<td></td>
</tr>
<tr>
<td>Study Goals</td>
<td>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 5. Successfully use software tools to analyse data and data processing</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures/Assignments</td>
</tr>
<tr>
<td>Expected Workload</td>
<td>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: 50 hours (10 × 5 hours)</td>
</tr>
<tr>
<td>Total = 140 hours</td>
<td></td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Will be provided online</td>
</tr>
<tr>
<td>Assessment</td>
<td>Course will be assessed by coursework. Students work in small groups on a coursework assignment for each topic. The coursework assignment will be marked and these marks will be combined into a single final mark for the course.</td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>NA</td>
</tr>
</tbody>
</table>
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. The proof assistant that we will be using is called Coq. As we will show during this course, proof assistants turn doing proofs and logic 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 an (imperative) programming language in logic.
- State and prove the correctness of program transformations.
- Use Hoare logic to prove properties of imperative programs.
- Use the Coq proof assistant to specify and prove correct a non-trivial program.

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

Main course material:

Software Foundations
Benjamin C. Pierce et al.
Freely available online at http://www.cis.upenn.edu/~bcpierce/sf/current/

Supplementary material:

Types and Programming Languages
Benjamin C. Pierce
MIT Press
ISBN 0-262-16209-1

Assessment

The final grade consists of the following parts:

- A programming project in the Coq 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. There is an oral resit for the written exam.

The research project and homework should be done individually.
### CS4180

**Deep Learning**

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. J.C. van Gemert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor</td>
<td>Dr. D.M.J. Tax</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/6/4</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Start Education</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Pattern Recognition, Statistics, Probability theory, Programming experience.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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 datasets and unstructured input data such as raw images/audio/text. Topics include: back-propagation, optimization, neural networks, convolutional nets, recurrent nets. The course will have lectures, a seminar and a lab practical: - The lectures will be on generic topics; building the backbone. - The seminar will have students read, critique, and present recent deep learning research papers. - The lab will have students apply a (small) deep learning project.</td>
</tr>
</tbody>
</table>
| 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. 
- [LO2]. Research literature concerning one of the above techniques, summarize it and report it to their peers (e.g., by means of a power-point presentation or a demonstration) 
- [LO3]. Debate upon positive and negative aspects of the techniques mentioned above  
- [LO4]. Implement one or more of the above mentioned techniques in a computer language and deep learning toolkit (e.g. Keras, TensorFlow, PyTorch, MatConvNet) 
- [LO5]. Determine which technique(s) is most appropriate for being used in a certain problem domain. 
- [LO6]. Apply the appropriate technique to a (simple) problem domain. |
| Education Method       | Lectures            |
| Literature and Study   | Book, freely available online: [http://www.deeplearningbook.org](http://www.deeplearningbook.org) |
| Materials              | Research papers that will be available through Brightspace. |
| Assessment             | 
  1. Presentation: during the seminar a small group of students presents a paper.  
  2. Students will have to submit *relevant* questions about the paper the day before.  
  3. Lab assignment: in a small group of students you experiment with one or more of the discussed techniques and write a research report, and you present your findings.  
  4. Exam about the papers and the theory. |
| Tags                   | Artificial intelligence, Image processing |

### CS4195

**Modeling and Data Analysis in Complex Networks**

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>H. Wang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/20</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Start Education</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>The assignment and final project require basic programming skill.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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 (co)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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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.</td>
</tr>
<tr>
<td>Education Method</td>
<td>In total, there will be 8 lectures (subject to minor changes) including a guest lecturer on applications to specific domains e.g. economy, social networks and the brain and a self-study lecturer. Students will also learn via an assignment and a final project (each group gets individual supervision).</td>
</tr>
<tr>
<td>Assessment</td>
<td>assignment and final project</td>
</tr>
<tr>
<td>CS4210-A</td>
<td>Algorithms for Intelligent Decision Making</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Responsible Instructor</td>
<td>Dr. N. Yorke-Smith</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. M.M. de Weerdt</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. M.T.J. Spaan</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/4/0 (i.e. 2 lectures/week in Q3)</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Start Education</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Recommended: IN4301: Advanced Algorithms, or equivalent; and/or IN4010: Artificial Intelligence Techniques, or equivalent</td>
</tr>
<tr>
<td>Required: basic course(s) in algorithm design and analysis, and complexity theory</td>
<td></td>
</tr>
<tr>
<td>Course Contents</td>
<td>Decision making is at the centre of artificial intelligence. This course gives you practical skills on a solid theoretical base:</td>
</tr>
<tr>
<td>* Modelling realistic combinatorial optimisation problems involving multiple users, and designing systems to solve such problems.</td>
<td></td>
</tr>
<tr>
<td>* Emphasis on problem characteristics seen in decision problems in energy, logistics, and health sectors.</td>
<td></td>
</tr>
<tr>
<td>* Mathematically-grounded techniques with computational feasibility: sequential decision making, algorithmic game theory, constraint programming.</td>
<td></td>
</tr>
<tr>
<td>Apply the skills you learn in this course by taking CS4210-B: Intelligent Decision Making Project in quarter 4!</td>
<td></td>
</tr>
<tr>
<td>Study Goals</td>
<td>By the end of this course, you will be able to identify features of real-world computational decision problems, and be able to model and design systems for simplified instances of these problems using partially-observable markov decision processes and constraint-based scheduling, taking into account rational, self-interested behaviour of participants.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures, homework exercises (optional), and programming assignments.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Provided on Brightspace</td>
</tr>
<tr>
<td>Assessment</td>
<td>The final mark depends on the marks obtained for</td>
</tr>
<tr>
<td>(a) programming assignments (3 in total) [60%] and</td>
<td></td>
</tr>
<tr>
<td>(b) the exam [40%].</td>
<td></td>
</tr>
<tr>
<td>The course contains 3 parts: Part 1, Part 2 and Part 3.</td>
<td></td>
</tr>
<tr>
<td>There are two optional homework exercises per part.</td>
<td></td>
</tr>
<tr>
<td>Homework exercises have to be completed individually unless otherwise stated.</td>
<td></td>
</tr>
<tr>
<td>There is a programming assignment to be completed at the end of each part. Each assignment is graded on a scale from 0 to 10.</td>
<td></td>
</tr>
<tr>
<td>The final mark for the programming assignment is a weighted average of the mark obtained for the three assignments.</td>
<td></td>
</tr>
<tr>
<td>Programming exercises can be completed by two students working together.</td>
<td></td>
</tr>
<tr>
<td>The exam is graded on a scale from 0 to 10.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Elective</td>
<td>Yes</td>
</tr>
<tr>
<td>Tags</td>
<td>Algorithms</td>
</tr>
<tr>
<td></td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td></td>
<td>Group work</td>
</tr>
<tr>
<td></td>
<td>Modelling</td>
</tr>
<tr>
<td></td>
<td>Optimisation</td>
</tr>
<tr>
<td></td>
<td>Programming</td>
</tr>
<tr>
<td></td>
<td>Projects</td>
</tr>
<tr>
<td></td>
<td>Small groups</td>
</tr>
</tbody>
</table>
| 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 probability 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 |
<p>| 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. |
| Assessment | Written exam + evaluation of the mini project |</p>
<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. P. Pawelczak</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/4/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Start Education</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
</tbody>
</table>

**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++). As for the latter, the suggestion is also to follow TI2726-B Embedded Software to improve C programming skills. Nonetheless, students with little knowledge of programming will be helped.

**Course Contents**  
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  
  - NB-IoT: protocol specifications, modulation format, network design

- **Review of wireless tools (example topics):**  
  - Introduction to wireless packet sniffing and analysis using Wireshark (assignment)  
  - Introduction to Software Defined Radio  
  - Simple simulations of WiFi network with NS3

- **RFID networking (example topics):**  
  - Principles of backscatter  
  - Protocol formats: EPC C1G2 and NFC-based formats  
  - 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.

**Practical Guide**  
For an introduction module on Software Defined Radio students are expected to get their own Software Defined Radio from http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/. Also, students should install https://www.gnuradio.org/ and/or https://nl.mathworks.com/hardware-support/rtl-sdr.html on their own laptops.

**Books**  

**Prerequisites**  
Background in C programming; prior knowledge of Matlab and Simulink; experience in Python programming is an advantage.

**Assessment**  
Points from the mini-project assignments and a written final exam. A research paper analysis from conferences such as IEEE INFOCOM, ACM MobiCom, ACM SIGCOMM will be required to pass the course.
<table>
<thead>
<tr>
<th><strong>ET4397IN</strong></th>
<th><strong>Network Security</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>C. Doerr</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/4/0</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Required for</strong></td>
<td>IN4253ET</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>The course provides an overview of the most important concepts, methods and best practices in computer and network security. Its goal is to equip students with the necessary background to understand the role and importance of security in the information society as well as to critically reflect upon and improve the security of computer and communication systems. 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.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>see course contents</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>Work sheets during the term and final exam. Final exam may be substituted with a hardware or software student project.</td>
</tr>
</tbody>
</table>
Course Contents

Have you ever wondered how Toy Story was made, why the game Uncharted 4 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... probably, 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 previulization 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.

Tentative Course Schedule:

1. Once upon a time - Introduction to Computer Graphics Get an overview of it all
2. Painting with numbers - Graphics Pipeline Understand how a graphics card works
3. It get's under my skin - Reflection Models Learn how light interacts with objects
4. Where there is light - Hard Shadows Discover how shadows are computed efficiently
5. The line begins to blur - Filtered Shadows Learn how to hide shadow artifacts and why signal theory relates to shadows
6. Be aware of your image - Image-Based Shadow Techniques Learn how to rely on image-like representations to compute shadows
7. The importance of being accurate - Visibility computations for shadows and radiosity Learn about physics and understand: A lot that we did was wrong!
8. Make it bouncy - Fast Global Illumination Learn about light interaction in scenes and how to tackle the recursivity of the rendering equation!
9. A Question of Time - Temporal Coherence Understand that redundancy can be exploited
10. Big is better - Volume Rendering with Exposure render - Learn about volume rendering and global illumination in volumes not relevant for the exam.
11. Its art! - Interaction and Design Get to know how movie makers control appearance
12. Express yourself - Non-photorealistic Rendering See that there is more than realism
13. I see double! - Stereovision Learn why Avatar's 3D is special
14. Let's talk about FX! - Article session Learn how to read (a scientific article)

There will also be practical sessions addressing several topics, such as 3D Modelling, Light Design, Mesh Simplification, and 3D Stereo rendering.

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, and 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 concepts within a graphics program.

Assessment

The final grade will be a weighted average of a paper presentation (25%), an oral exam covering the course content (25%), which might take the form of a (multiple-choice) exam (only pen and paper permitted), and a group project (50%), including a presentation and questions. The course is passed if this weighted average is >=6.

The estimated workload is roughly distributed as: 50% lectures (exam preparation), 40% project development, 10% paper presentation. The reason for the differing percentages with respect to their weighting is that the lectures form the basis of the exam but are also relevant for the project and paper presentation.
**IN4179 Intelligent User Experience Engineering**

<table>
<thead>
<tr>
<th><strong>Responsible Instructor</strong></th>
<th>Prof.dr. M.A. Neerincx</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Co-responsible for assignments</strong></td>
<td>R.J.L. Boumans</td>
</tr>
<tr>
<td><strong>Co-responsible for assignments</strong></td>
<td>M.E.U. Ligthart</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/2/2</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>none</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
</tbody>
</table>
| **Course Contents** | Whether you are playing a game in virtual reality, driving a semi-autonomous car, educating yourself in a MOOC, or controlling your smart home system from work; nowadays intelligent networked information and communication technology is omnipresent. A major design challenge is how the user experiences created with these technologies can be shaped such that they are, e.g., effective, efficient, enjoyable, trustworthy, respectful, satisfying, usable, confidential, supportive, and learnable. In the Intelligent User Experience Engineering (IUXE) course (MSc level), you will become acquainted with the application of a coherent cognitive engineering methodology for the design and evaluation of human–agent collaboration. Based on this methodology, we will elaborate on the state of the art of intelligent user interfaces (ePartners), such as artificial personal assistants, behavior change support systems, artificial team mates, eCoaches, social intelligence, and companion agents. The main topics of study are: Design methods: Situated Cognitive Engineering, Value Sensitive Design, Scenario-based Design, Claims Analysis, Design Rationale, Interaction Design Patterns. Conceptual Design: Knowledge Representation, Ontology Engineering, ePartners, Adaptive Automation, Socially Intelligent Interfaces. Design Evaluation: Prototyping, Test Methods, Measures, Questionnaires, Ethics. Human Factors Theories and Models: Human Cognition & Learning, Task Load, Emotion, Human–Agent Teamwork, Behavior Change and Persuasive Technology. The module is assessed by paper writing, a presentation and course (project) work of the group, as follows: 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 user-based 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 cycle, with sufficient detail for a new group of researchers to continue the research. 10. Write a short scientific paper (extended abstract) for a conference. 11. Present work on a design project to an academic audience. 12. 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 intelligent user experience engineering. Students are required to read a number of scientific papers which are made available on Blackboard, 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, implementation, and evaluation) for a given design problem. This year (like the past years), the design problem is a social robot, either for children with a disease or older adults with dementia, and their social environment. The objective of the social robot is to improve users physical, social, cognitive, and emotional well-being. The students will use the situated Cognitive Engineering Tool (sCET) to specify the design rationale and its evaluation. Throughout the course, students will give presentations about their progress on a regular basis during the lectures. In total there are 3 presentations and one final report (so-called milestones). Each student will give one presentation. **Literature and Study Materials** Papers from scientific journals on Blackboard. Lecture notes on Blackboard. **Assessment** The module is assessed by paper writing, a presentation and course (project) work of the group, as follows: Short scientific paper on the project that summarizes the design rationale (relating the research to recent literature) and evaluation results (40%). Presentation on a project milestone during the course (20%) Project report according to the prescribed format (40%) **Exam Hours** There is no exam. The assessment is based on a paper, presentation and (group) report. During the course, students will receive feedback on interim work. There is no resit after the end of the course. **Enrolment / Application** Basic prior knowledge on human-computer interaction is helpful, but not required. Enrolment should be done via Brightspace.
### 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.

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, final presentation of result, presentation of ongoing project progress, participation in discussions, overall quality of the practical work and class attendance. 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.

### maximum aantal deelnemers

If there is an unexpected high demand for this course, then enrollment will be based on past performance in relevant courses.
### Software Architecture

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof.dr. A. van Deursen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/4/0</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>Different, to be announced</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Software engineering</td>
</tr>
</tbody>
</table>

**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 recognise technical debt and have an understanding of possible refactoring.

**Education Method**

Interactive lectures, lab assignment, paper presentation and discussion.

**Literature and Study Materials**


**Assessment**

No written exams. Lab assignment, paper and presentation.

**Special Information**

Twitter handle: https://twitter.com/delftswa

**Co-Instructor**

M. Finavaro Aniche

---

### Machine learning

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>M. Loog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor</td>
<td>Dr. D.M.J. Tax</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. J.C. van Gemert</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr.ing. J. Kober</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/2/2</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>4</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>IN4085</td>
</tr>
</tbody>
</table>

**Course Contents**

The course will treat a number of machine learning topics, approaches, and techniques in detail and on an advanced level. Possible topics are:

- learning theory
- complexity
- semi-supervised learning
- multiple instance learning
- kernel methods
- reinforcement learning
- Gaussian processes
- sparsity

**Study Goals**

After the course, the student is able to recognize the (limits to the) practical applicability of the presented theory. Moreover, s/he is able to see the relationships of a novel technique to those discussed in the course, and has insight in what type of problem requires application of which type of machine learning technique.

**Education Method**

We follow a scheme in which every topic is treated in a two-week block. In the first week, one of lecturer will present a technique based on a tutorial paper or other reading material. In the second week, the student will work on an exercise that extends and deepens their knowledge and understanding of the technique under consideration in that particular block. A part of the exercises involves programming. The final output to every exercise is a short report covering the necessary derivations, explanations, general text, figures, possibly snippets of code, etc.

**Literature and Study Materials**

For every block of two weeks, in which a single topic is treated, specific literature will be provided through Brightspace.

**Assessment**

Grading is based on the six reports handed in (60-80%) and the final assignment (about 40-20%), the latter of which is based on a somewhat larger and more advanced machine learning challenge that the students will write a report on as well. There is no resit; not overall, nor for any of the elements.
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 objective of the IN4325 - Information Retrieval course is to introduce the scientific underpinnings of the field of Information Retrieval. 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;
- Natural language processing;
- Natural language generation

Study Goals

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

- Describe the different information retrieval models, and compare their strengths 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]
- Design and develop NLP applications. [Learning Objective 7]
- Evaluate NLP applications in different task contexts. [Learning Objective 8]
- Illustrate suitable application scenarios for advanced Natural language processing topics such as natural language queries and summarization. [Learning Objective 9]

Expected workload is 140 hours: 45 hours for lectures and lecture preparation plus the weekly assignment, 65 hours for the group project and 30 hours for the literature survey.

The group assignment is performed collectively, but graded individually. Assignments have no resit opportunities.
Responsible Instructor: Dr. M. Nasri Nasrabadi

Contact Hours / Week: 0/0/4/0 and 0/0/4/0 lab

Education Period: 3

Start Education: 3

Exam Period: 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
- Resource access policies in real-time systems
- 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)


Assessment: Written exam (grade) + lab work (grade); the exam has a resit

Exam Hours: 3

Permitted Materials during Tests: Simple calculator

Co-Instructor: Prof. Dr. K.G. Langendoen
<table>
<thead>
<tr>
<th>IN4391</th>
<th>Distributed Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Dr. J.S. Rellermeyer</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/2/0</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>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. Topics discussed in class include:</td>
</tr>
<tr>
<td></td>
<td>- communication, messaging, naming</td>
</tr>
<tr>
<td></td>
<td>- scheduling of distributed workloads, resource provisioning</td>
</tr>
<tr>
<td></td>
<td>- fault-tolerance, reliability, availability, fault-tolerant protocols</td>
</tr>
<tr>
<td></td>
<td>- consistency, replication, distributed file systems</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>1. Explain the objectives and functions of distributed computing systems.</td>
</tr>
<tr>
<td></td>
<td>2. Describe how distributed computing systems have evolved, over time, from primitive batch systems to sophisticated multi-user systems.</td>
</tr>
<tr>
<td></td>
<td>3. Describe the architecture and operation of distributed computing systems.</td>
</tr>
<tr>
<td></td>
<td>4. Explain how distributed computing systems can process user workloads.</td>
</tr>
<tr>
<td></td>
<td>5. Explain how distributed computing systems can detect and correct faults and errors.</td>
</tr>
<tr>
<td></td>
<td>6. Implement complex operations of modern distributed computing systems in realistic scenarios.</td>
</tr>
<tr>
<td></td>
<td>7. Analyze the trade-offs inherent in the design of distributed computing systems (performance, efficiency, scalability, reliability, availability, fault-tolerance.)</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Lectures: 7 weeks x 2h</td>
</tr>
<tr>
<td></td>
<td>Additional material: Several relevant research articles introduce the student to the latest advances on the topic.</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>Overview: The course consists of two main parts: the theoretical Part I and the practical Part II. Students are assessed through a written exam (open questions) and a demonstration of practical skills.</td>
</tr>
<tr>
<td></td>
<td>The result of Part I of this course must be at least 6. The result of Part II of this course (practical) must be Completed (C, Voltooid/V in Dutch). The final grade is the result of Part I.</td>
</tr>
</tbody>
</table>
IN4393  Computer Vision

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. J.C. van Gemert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/2/0/ lecture + 0/0/2/0 seminar + 0/0/4/0 lab</td>
</tr>
<tr>
<td>Education Period</td>
<td>3</td>
</tr>
<tr>
<td>Start Education</td>
<td>3</td>
</tr>
<tr>
<td>Exam Period</td>
<td>3</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>You are expected to have a working understanding of image processing, linear algebra, and of probability and statistics. Knowledge about pattern recognition and/or machine learning is preferred. The parallel course “Deep Learning” will combine well with Computer Vision.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>The central theme of the computer vision course is the automatic analysis and interpretation of images and videos using computer algorithms. The course explores a range of techniques for image analysis, image matching, image stitching, 3D reconstruction.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>After successfully completing this course:</td>
</tr>
<tr>
<td></td>
<td>- You are able to explain and implement various techniques for feature point detection, and can explain the type of feature points these detectors identify.</td>
</tr>
<tr>
<td></td>
<td>- You are able to explain and implement techniques for feature point description and feature point matching. You are able to use these techniques in applications such as object detection and image stitching.</td>
</tr>
<tr>
<td></td>
<td>- You are able to explain and implement techniques for image stitching. The student understands the key problems in developing image-stitching algorithms.</td>
</tr>
<tr>
<td></td>
<td>- You are able to explain and implement basic techniques for feature tracking.</td>
</tr>
<tr>
<td></td>
<td>- You are able to develop and explain computer vision systems for real-world applications. In particular, you are able to select computer vision techniques that are to solve a specific image analysis or image understanding problem, to motivate this selection, and to combine the selected techniques into a working computer vision system.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures</td>
</tr>
<tr>
<td></td>
<td>Lab project.</td>
</tr>
<tr>
<td></td>
<td>Seminar: paper reading, critiquing, and presenting.</td>
</tr>
<tr>
<td>Assessment</td>
<td>1. Presentation: during the seminar a small group of students presents a paper.</td>
</tr>
<tr>
<td></td>
<td>2. The groups that are not presenting have to submit “relevant” questions about the paper the day before.</td>
</tr>
<tr>
<td></td>
<td>3. Lab assignment: in a small group of students you experiment with one or more of the discussed techniques and write a research report, and you present your findings.</td>
</tr>
<tr>
<td></td>
<td>4. Participation in class</td>
</tr>
<tr>
<td></td>
<td>5. Exam about the papers and the theory.</td>
</tr>
<tr>
<td>Tags</td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td></td>
<td>Image processing</td>
</tr>
<tr>
<td>Year</td>
<td>2018/2019</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Organization</td>
<td>Electrical Engineering, Mathematics and Computer Science</td>
</tr>
<tr>
<td>Education</td>
<td>Master Computer Science</td>
</tr>
</tbody>
</table>

**Specialistievakken start kwartaal 4 2018**
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 five such techniques:

- behavioral profiling and anomaly detection;
- data stream mining and distributed data processing;
- web-crawling and text mining;
- software fuzzing and protocol reverse-engineering; and
- information fusion and collaborative knowledge discovery.

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 handle the different kinds of input data used to construct these profiles such as websites and software logs. In addition to the traditional sample data sets, software code and implementations form an important source of information for cyber data analytics. In addition to training from execution logs, the student will learn how to use this information source by actively providing input and learning from the returned output.

The student will be able to:

- Develop and analyze algorithms that learn models from large data streams;
- Detect anomalies in system logs, e.g., for fraud detection;
- Construct behavioral profiles of both people and software;
- Learn insightful models from multiple data sources (e.g., websites, network traces, software code);
- Apply knowledge fusion and collaborative knowledge discovery methods;
- Use machine learning to discover and analyze threats in software components.

There will be two lectures for each of the five topics, and a large lab exercise in which teams of two students will work on a use-case of one of these topics. Each team is free to choose their own topic from a selection of recent research in cyber data analytics.

One large lab assignment in teams of two students resulting in a written report (50%) and an individual summative exam on selected content (50%).
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.

After three weeks of core topics, the course offers two specialization tracks:

- **MMSR Analytics**, focusing on data analytics aspects for multimedia search and recommendation with special focus on emerging topics.
- **MMSR Systems**, focusing on system and implementation aspects for multimedia search and recommendation with special 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;
- 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;
- explain the difference between topical relevance and utility in multimedia search and recommendation.

In addition to the core goals, students choosing the MMSR Analytics specialization will be able to:
- describe and implement cross-disciplinary approaches to multimedia search and 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.

In addition to the core goals, students choosing the MMSR Systems specialization will be able to:
- describe and implement practical solutions to deal with real-world multimedia search and/or recommendation;
- 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.

### Special Information

Please see the Brightspace pages of this course for further information about course organization and suggested prerequisite knowledge.
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Embedded Systems Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Prof. dr. K.G. Langendoen</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/0/4 &amp; lab</td>
</tr>
<tr>
<td>Education Period</td>
<td>4</td>
</tr>
<tr>
<td>Start Education</td>
<td>4</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>MUST have C programming skills. Students who have taken the TI2726-B Emb. Software course automatically qualify, others will have to pass an on-line ACCEPTANCE test.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>This is a core course of the Masters in Embedded Systems.</td>
</tr>
<tr>
<td>Education Method</td>
<td>Lectures (8<em>2hrs), lab work (8</em>4hrs), coding@home (8*12hrs), report (8hrs), so on average 2 days per week</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Lecture notes + Website</td>
</tr>
<tr>
<td>Assessment</td>
<td>Lab. project (75%) + written report (25%), no exam, no resit</td>
</tr>
<tr>
<td>Enrolment / Application</td>
<td>The capacity is limited and -as this is a compulsory course for ES students- they get preference over other MSc students.</td>
</tr>
</tbody>
</table>
| 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 algorithmically 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 fulfill their data needs, both by involving employees, and by reaching out to anonymous crowds through online marketplaces like Amazon Mechanical Turk and CrowdFlower.

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, semi-autonomous systems, citizen science, crowdsourcing, and human computer 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

This course consists of 16 2-hour lectures.

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 preparation, 24 hours for preparing final survey, and 5 hours for exam and plenary presentations (total 140 hours).

Books:
Assessment

The final grade consists of the following parts:

- Weekly Individual assignment, weighting 10% of the final grade
- Group assignment, weighting 60% 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.

Tags

Algorithmics  Artificial intelligence  Design  Programming  Software

<table>
<thead>
<tr>
<th>CS4205</th>
<th>Evolutionary Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Instructor</td>
<td>Prof. dr. P.A.N. Bosman</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/0/4</td>
</tr>
<tr>
<td>Education Period</td>
<td>4</td>
</tr>
<tr>
<td>Start Education</td>
<td>4</td>
</tr>
<tr>
<td>Exam Period</td>
<td>4</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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 design super 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, linkage learning, multi-modal optimization, multi- and many-objective optimization, and real-world applications.

The course will have 9 lectures and a lab practical. The lab practical will exist of 2 parts. The first part pertains to experimenting with already implemented EAs on predefined problems. In the second part, you will build (in a group) your own EA to evolve a solution to a problem of your choosing. Depending on the number of students that have enrolled, the final lecture will be used to have students present their results.

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 swarm-based 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

9 Lectures
2 Lab projects

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.

Permitted Materials during Tests

None

Tags

Algorithmics  Artificial intelligence  Optimisation
## Intelligent Decision Making Project

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. M.T.J. Spaan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor</td>
<td>Dr. N. Yorke-Smith</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. M.M. de Weerdt</td>
</tr>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/0/1</td>
</tr>
<tr>
<td>Education Period</td>
<td>4</td>
</tr>
<tr>
<td>Start Education</td>
<td>4</td>
</tr>
<tr>
<td>Exam Period</td>
<td>4</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>CS4210-A: Algorithms for Intelligent Decision Making</td>
</tr>
<tr>
<td>Course Contents</td>
<td>Decision making is at the centre of artificial intelligence. Building upon theoretical knowledge gained in the course CS4210-A Algorithms for Intelligent Decision Making, 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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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.</td>
</tr>
<tr>
<td>Education Method</td>
<td>A research project in a small group.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Mainly survey papers and book chapters. Details are provided via Brightspace.</td>
</tr>
<tr>
<td>Assessment</td>
<td>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 (20%) 4. Oral presentation of the research project (20%) Only items 1 and 2 can be examined a second time.</td>
</tr>
<tr>
<td>Enrolment / Application</td>
<td>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.</td>
</tr>
<tr>
<td>Tags</td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td>maximum aantal deelnemers</td>
<td>30</td>
</tr>
</tbody>
</table>

## Error Correcting Codes

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr.ir. J.H. Weber</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/0/3</td>
</tr>
<tr>
<td>Education Period</td>
<td>4</td>
</tr>
<tr>
<td>Start Education</td>
<td>4</td>
</tr>
<tr>
<td>Exam Period</td>
<td>4</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>A B.Sc. Programme in Electrical Engineering, Computer Science, or Mathematics</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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. This can be done in various ways. * &quot;Broad&quot;: The student solves 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. * &quot;Narrow, but in-depth&quot;: In consultation with the lecturer, the student chooses a certain topic from the course, which is investigated in more detail, by writing either an essay (discussing a paper from the recent literature) or a computer program for a demo explaining the chosen topic (to be used by the lecturer in class room).</td>
</tr>
<tr>
<td>Education Method</td>
<td>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.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>Lecture notes &quot;Error-Correcting Codes&quot; by J.H. Weber</td>
</tr>
<tr>
<td>Assessment</td>
<td>Written (closed book), oral (closed book), essay, or computer program</td>
</tr>
<tr>
<td>Remarks</td>
<td>Actual course information available on Brightspace.</td>
</tr>
</tbody>
</table>
### ET4285  Measuring and Simulating the Internet

<table>
<thead>
<tr>
<th><strong>Responsible Instructor</strong></th>
<th>Dr. ir. F.A. Kuipers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/0/2</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>none</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>(Advanced) Networking course (e.g., CS4055) and Programming skills.</td>
</tr>
</tbody>
</table>

**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.

**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.

**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.

### IN4182  Digital Audio and Speech Processing

<table>
<thead>
<tr>
<th><strong>Responsible Instructor</strong></th>
<th>Prof. dr. ir. R. Heusdens</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/0/4</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>none</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>Signal processing, stochastic processes and preferably statistical signal processing. Knowledge of MATLAB is advantageous.</td>
</tr>
</tbody>
</table>

**Course Contents**
In the course Audio and Speech Processing, attention will be paid to speech and audio processing algorithms. Speech and audio processing algorithms are applied in many applications, like mobile voice communication and storage/streaming of audio/speech. In the course the following topics will be considered: short-time Fourier transform based noise suppression, optimal minimum mean-squared error estimators, tracking of noise power spectral density, speech production model, linear prediction, speech coding, psychoacoustics, spectral/temporal masking, perceptual audio coding (sinusoidal coding, waveform coding), MPEG audio coding standards (incl. mp3), multi-channel audio coding.

**Study Goals**
1. Audio coding
   - Keywords: psycho-acoustics, spectral/temporal masking, time-to-frequency transformations, audio coding standards, multi-channel coding
   - The student is able to explain the general mechanisms of human auditory perception
   - is able to discuss the use of human auditory perception in state-of-the-art audio compression algorithms
   - is able to discuss the structure and principles used in standardized perceptual audio coding algorithms
   - 2. Speech enhancement
   - keywords: spectral subtraction, Wiener filtering, conditional mean estimators, noise power spectral density tracking
   - The student is able to implement state-of-the-art algorithms for suppressing noise in noisy speech signals
   - is able to design components/functional blocks in noise suppression algorithms (e.g. derive Bayesian suppression rules)
   - 3. Speech coding
   - keywords: speech production, linear predictive analysis, fundamental frequency estimation, voicing estimation, speech coding based on linear prediction
   - The student is able to identify how the human speech production process is exploited in speech processing algorithms
   - is able to implement a linear predictive speech synthesizer
   - is able to discuss the signal processing techniques used in speech coding

**Education Method**
Lectures, mini projects + hands-on exercises

**Literature and Study Materials**
Overhead sheets + selected articles/book chapters

**Assessment**
Oral examination + evaluation of the mini projects

**Special Information**
Participation in the mini projects is mandatory to qualify for the exam.
<table>
<thead>
<tr>
<th><strong>IN4185</strong></th>
<th><strong>Globally Distributed Software Engineering</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>Prof.dr.ir. D.M. van Solingen</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>colleges en practicum 0/0/0/4</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>Exam by appointment</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>Software Engineering (= IN2705)</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>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 EDX with international participants as well. Lectures and excercises are followed digitally. The course hours in the calendar are used for interaction with the professor and more detailed discussion and feedback.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>Digital lectures, home work assignments and distributed group assignment.</td>
</tr>
<tr>
<td><strong>Computer Use</strong></td>
<td>The course does not contain programming exercises. 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.</td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
<td>Presentation handouts</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>Written report on lab work and literature research</td>
</tr>
<tr>
<td><strong>Enrolment / Application</strong></td>
<td>Please enroll</td>
</tr>
<tr>
<td><strong>Special Information</strong></td>
<td>Please contact <a href="mailto:d.m.vansolingen@tudelft.nl">d.m.vansolingen@tudelft.nl</a></td>
</tr>
<tr>
<td><strong>IN4254</strong></td>
<td><strong>Smart Phone Sensing</strong></td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td><strong>Responsible Instructor</strong></td>
<td>M.A. Zuñiga Zamalloa</td>
</tr>
<tr>
<td><strong>Contact Hours / Week</strong></td>
<td>0/0/2</td>
</tr>
<tr>
<td><strong>Education Period</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Start Education</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Exam Period</strong></td>
<td>none</td>
</tr>
<tr>
<td><strong>Course Language</strong></td>
<td>English</td>
</tr>
<tr>
<td><strong>Expected prior knowledge</strong></td>
<td>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 &quot;Object Oriented Programming&quot; 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 &quot;Probability and Statistics&quot;.</td>
</tr>
<tr>
<td><strong>Course Contents</strong></td>
<td>We will be refreshing some concepts on Probability, but we will not be refreshing concepts on Object Oriented Programming. 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.</td>
</tr>
<tr>
<td><strong>Study Goals</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Education Method</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Literature and Study Materials</strong></td>
<td>Research Papers and web tutorials</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>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 (70% of grade) 4) Oral exam (10% 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.</td>
</tr>
<tr>
<td><strong>Enrolment / Application</strong></td>
<td>1. You need to enrol in Brightspace 2. The first lecture will be compulsory 3. Only a limited number of students may be able to take the course (due to the capacity of facilities). If this occurs, priority will be given to students that register first and that satisfy the necessary prerequisites.</td>
</tr>
</tbody>
</table>
### IN4255  Geometric Data Processing

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr. K.A. Hildebrandt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/0/4</td>
</tr>
<tr>
<td>Education Period</td>
<td>4</td>
</tr>
<tr>
<td>Start Education</td>
<td>4</td>
</tr>
<tr>
<td>Exam Period</td>
<td>4</td>
</tr>
<tr>
<td>Education Period</td>
<td>5</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Basic knowledge in mathematics (linear algebra, calculus): TH1106M, TH1206M or comparable courses. Students who haven't followed any of these courses can follow the course, but should be willing to invest more time.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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</td>
</tr>
<tr>
<td>Education Method</td>
<td>The course will combine lectures, tutorials, practical project work, and homework assignments.</td>
</tr>
<tr>
<td>Assessment</td>
<td>To pass the course, the practical projects need to be completed successfully and a final exam must be passed. The homework assignments help with preparing for the exam. The final grade is the mean of the grade of the final exam and the grade of the project work. A resit is possible only for the final exam.</td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>For the exam, students are allowed to bring two pages of hand-written notes.</td>
</tr>
</tbody>
</table>

### IN4306  Literature Survey

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Dr.ir. J.F.M. Tonino</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Education Period</td>
<td>None (Self Study)</td>
</tr>
<tr>
<td>Start Education</td>
<td>1</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Course Contents</td>
<td>The Literature Survey is an individual assignment carried out under supervision of a CS 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.</td>
</tr>
<tr>
<td>Study Goals</td>
<td>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 specialisation. 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, ...).</td>
</tr>
<tr>
<td>Education Method</td>
<td>Individual assignment and individual guidance by a scientific staff member.</td>
</tr>
<tr>
<td>Assessment</td>
<td>Writing a scientific report, individually and under supervision of a staff member. This staff member will also mark the report.</td>
</tr>
<tr>
<td>Enrolment / Application</td>
<td>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.</td>
</tr>
<tr>
<td>Co-Instructor</td>
<td>Dr. C. Lofi</td>
</tr>
<tr>
<td>Co-Instructor</td>
<td>Dr.ir. A.R. Bidarra</td>
</tr>
<tr>
<td>Co-Instructor</td>
<td>Prof.dr. S.T. Erdweg</td>
</tr>
<tr>
<td>Co-Instructor</td>
<td>C. Doerr</td>
</tr>
<tr>
<td>Co-Instructor</td>
<td>Dr. K.A. Hildebrandt</td>
</tr>
</tbody>
</table>
### IN4333: Language Engineering Project

<table>
<thead>
<tr>
<th>Responsible Instructor</th>
<th>Prof.dr. E. Visser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Hours / Week</td>
<td>0/0/0/4</td>
</tr>
<tr>
<td>x/x/x/x</td>
<td></td>
</tr>
<tr>
<td>Education Period</td>
<td>4</td>
</tr>
<tr>
<td>Start Education</td>
<td>4</td>
</tr>
<tr>
<td>Exam Period</td>
<td>none</td>
</tr>
<tr>
<td>Course Language</td>
<td>English</td>
</tr>
<tr>
<td>Expected prior knowledge</td>
<td>Students are expected to have completed (the lab of) the compiler construction course IN4303.</td>
</tr>
<tr>
<td>Course Contents</td>
<td>&quot;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 obscures the concepts under consideration. Linguistic abstraction can be used as a tool to capture our emerging understanding of domains of computation.’ (Visser, SCP 2014)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 <em>before</em> the course starts. 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.</td>
</tr>
<tr>
<td>Education Method</td>
<td>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.</td>
</tr>
<tr>
<td>Literature and Study Materials</td>
<td>- Documentation of the design and implementation of a specific language - Papers about language definition techniques</td>
</tr>
<tr>
<td>Books</td>
<td>See Literature and Study Materials</td>
</tr>
<tr>
<td>Assessment</td>
<td>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.</td>
</tr>
<tr>
<td>Permitted Materials during Tests</td>
<td>not applicable</td>
</tr>
<tr>
<td>Judgement</td>
<td>The final grade is based on the following components: - github repository with language project (40%) - written report about the project (30%) - presentation (slides) (30%)</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Prof.dr.ir. K.I. Aardal</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof.dr. P.A.N. Bosman</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>R.J.L. Boumans</td>
<td></td>
</tr>
<tr>
<td>Prof.dr.ir. A. Bozzon</td>
<td>Web Information Systems</td>
</tr>
<tr>
<td></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr.ir. W.P. Brinkman</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr.ing. D.J. Broekens</td>
<td></td>
</tr>
<tr>
<td>Dr. A.J. Cabo</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. L. di Carlo</td>
<td>QuTech</td>
</tr>
<tr>
<td></td>
<td>Technische Natuurwetenschappen</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>P.S. Cesar Garcia</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof.dr. A. van Deursen</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>C. Doerr</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>R. Eggermont</td>
<td>Universiteitdienst</td>
</tr>
<tr>
<td>Dr. D. Elkouss Coronas</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof.dr. S.T. Erdweg</td>
<td>Programming Languages</td>
</tr>
<tr>
<td>Dr. J.C. van Gemert</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Dr. ir. G. Gousios</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof. dr. A. Hanjalic</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. C. Hauff</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. ir. R.C. Hendriks</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. ir. F.F.J. Hermans</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof. dr. ir. R. Heusdens</td>
<td></td>
</tr>
<tr>
<td>Dr. K.A. Hildebrandt</td>
<td></td>
</tr>
<tr>
<td>R.N.A. Holwerda</td>
<td></td>
</tr>
<tr>
<td>Prof. dr. ir. G.J.P.M. Houben</td>
<td></td>
</tr>
<tr>
<td>Dr. H.S. Hung</td>
<td></td>
</tr>
<tr>
<td>Prof. dr. C.M. Jonker</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Dr. A. Katsifodimos</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr.ir. J.J.A. Keiren</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. R.J. Krebbers</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr.ir. F.A. Kuipers</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof.dr. K.G. Langendoen</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. C. Lofi</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. M. Nasri Nasrabadi</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td><strong>Prof.dr. M.A. Neerinx</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr. F.A. Oliehoek</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr. P. Pawelczak</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr.ir. J.A. Pouwelse</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr. J.S. Rellermeyer</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr. B. Rieger</strong></td>
<td>Technische Natuurwetenschappen</td>
</tr>
<tr>
<td><strong>Prof.dr.ir. D.M. van Solingen</strong></td>
<td>Software Engineering</td>
</tr>
<tr>
<td><strong>Dr. M.T.J. Spaan</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr. D.M.J. Tax</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td><strong>Dr. N. Tintarev</strong></td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Dr. ir. J.F.M. Tonino</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>R.R. Venkatesha Prasad</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Prof. dr. E. Visser</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. F.M. Vos</td>
<td>Technische Natuurwetenschappen</td>
</tr>
<tr>
<td>Dr. M.M. de Weerdt</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Prof. dr. S.D.C. Wehner</td>
<td>QuTech</td>
</tr>
<tr>
<td>Dr. N. Yorke-Smith</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>Dr. A.E. Zaidman</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
<tr>
<td>M.A. Zuñiga Zamalloa</td>
<td>Elektrotechn., Wisk. &amp; Inform.</td>
</tr>
</tbody>
</table>

ontbreekt

Drs. M.A. van Loo