Maastricht University

Faculty of Science and Engineering

Department of Data Science and Knowledge Engineering (DKE)
# Contents

## Academic Calendar 2021 - 2022

1. **Education: The Profile of the Bachelor in Data Science and Artificial Intelligence**
   - 1.1 What is the programme about
   - 1.2 Study System
   - 1.3 Study Abroad
   - 1.4 Degree
   - 1.5 Study Feasibility and Quality Assurance
   - 1.6 Courses at other faculties or universities

2. **Bachelor**
   - 2.1 Curriculum of the First Year of the Bachelor’s Programme
   - 2.2 Curriculum of the Second Year of the Bachelor’s Programme
   - 2.3 Curriculum of the Third Year of the Bachelor’s Programme
   - 2.4 DKE Honours programme

3. **Master**
   - 3.1 Curriculum of the First Year of the Master's Programmes
   - 3.2 Year 2 of the Master's Programmes AI and DSDM
   - 3.3 Master's thesis AI & DSDM (Code: KEN4160 & Code: KEN4260)

4. **Facilities for Students**
   - 4.1 Student Affairs Office
   - 4.2 Administrative structure of the Faculty
   - 4.3 Teaching Material
   - 4.4 Participation in the Education
   - 4.5 Announcements concerning Educational Matters
   - 4.6 Change of Address Student
   - 4.7 Project rooms
   - 4.9 Student Services Centre (SSC)
   - 4.10 Education and Examination Regulations 2021-2022 (EER)

5. **Staff**
   - 5.1 Academic Staff
   - 5.2 Lab assistants
   - 5.3 Support Staff
Academic Calendar 2021 - 2022

Bachelor's Programme Data Science and Artificial Intelligence (year 1, 2, 3) and Master's Programmes Artificial Intelligence and Data Science for Decision Making- (year 1, 2)

Inkom Maastricht University
August 2021

Education periods
Period 1: 30 August - 15 October 2021
Period 2: 25 October - 10 December 2021
Period 3: 3 January 2022 - 21 January 2022
Period 4: 31 January - 25 March 2022
Period 5: 4 April - 25 May 2022
Period 6: 7 June - 24 June 2022

Project weeks
Period 3: Project weeks 3 - 21 January 2022
BA year 1, Final presentation: January 2022
BA year 2, Final presentation: January 2022
BA year 3, Final presentation: January 2022
MA year 1, Project seminar: January 2022

Period 6: Project weeks 7 - 24 June 2022
BA year 1, Final presentation: June 2022
BA year 2, Final presentation: June 2022
MA year 1, Project seminar: June 2022

BA Thesis Winter Conference: December 2021
BA Thesis Summer Conference: June 2022

Introduction day
23 & 24 August 2021 (Bachelor) and 26 & 27 August 2021 (Masters)

Exam and Resit periods
Period 1 - Exams: 18 - 22 October 2021
Period 2 - Exams: 13 - 17 December 2021
Period 3 - Resits semester 1: 24 - 28 January 2022
Period 4 - Exams: 28 March - 1 April 2022
Period 5 - Exams: 30 May - 3 June 2022
Period 6 - Resits semester 2: 27 June - 1 July 2022
*Please note, resits of bachelor year 3 period 4 take place in the exam week of period 5 - Exams: 30 May - 3 June 2022

Graduation
T.b.d. (Bachelor)
T.b.d. (Masters)

/Public) Holiday, no courses
Christmas: 20 December 2021 - 2 January 2022
New Year’s Day: 1 January 2022
Carnival break: 28 – 6 March 2022
Good Friday: 15 April 2022
Easter Monday: 18 April 2022
Dutch King’s Day: 27 April 2022
Liberation Day & Bridging Day: 5 & 6 May 2022
Ascension Day & Bridging Day: 26 & 27 May 2022
Whit Monday: June 6 2022

DKE students will have to register for their own courses, exams and resits through the Student Portal.
Note: exams can also be scheduled in the evening, from 06.00 p.m. until 09.00 p.m.
<table>
<thead>
<tr>
<th>August</th>
<th>September</th>
<th>October</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1</td>
<td>1 2 3 4 5</td>
<td>1</td>
</tr>
<tr>
<td>2 3 4 5 6 7 8</td>
<td>6 7 8 9 10 11 12</td>
<td>4 5 6 7 8 9 10 11</td>
</tr>
<tr>
<td>9 10 11 12 13 14 15</td>
<td>13 14 15 16 17 18 19</td>
<td>11 12 13 14 15 16 17</td>
</tr>
<tr>
<td>16 17 18 19 20 21 22</td>
<td>20 21 22 23 24 25 26</td>
<td>18 19 20 21 22 23 24</td>
</tr>
<tr>
<td>23 24 25 26 27 28 29</td>
<td>27 28 29 30</td>
<td>25 26 27 28 29 30</td>
</tr>
<tr>
<td>30 31</td>
<td></td>
<td>30 31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>November</th>
<th>December</th>
<th>January</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7</td>
<td>1 2 3 4 5</td>
<td>1</td>
</tr>
<tr>
<td>8 9 10 11 12 13 14</td>
<td>6 7 8 9 10 11 12</td>
<td>3 4 5 6 7 8 9</td>
</tr>
<tr>
<td>15 16 17 18 19 20 21</td>
<td>13 14 15 16 17 18 19</td>
<td>10 11 12 13 14 15 16</td>
</tr>
<tr>
<td>22 23 24 25 26 27 28</td>
<td>20 21 22 23 24 25 26</td>
<td>17 18 19 20 21 22 23</td>
</tr>
<tr>
<td>29 30</td>
<td>27 28 29 30 31</td>
<td>24 25 26 27 28 29 30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>February</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7</td>
<td>1 2 3 4 5</td>
<td>1</td>
</tr>
<tr>
<td>7 8 9 10 11 12 13</td>
<td>7 8 9 10 11 12 13</td>
<td>4 5 6 7 8 9 10</td>
</tr>
<tr>
<td>14 15 16 17 18 19 20</td>
<td>14 15 16 17 18 19 20</td>
<td>11 12 13 14 15 16 17</td>
</tr>
<tr>
<td>21 22 23 24 25 26 27</td>
<td>21 22 23 24 25 26 27</td>
<td>18 19 20 21 22 23 24</td>
</tr>
<tr>
<td>28</td>
<td>28 29 30 31</td>
<td>25 26 27 28 29 30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>May</th>
<th>June</th>
<th>July</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1</td>
<td>1 2 3 4 5</td>
<td>1</td>
</tr>
<tr>
<td>2 3 4 5 6 7 8</td>
<td>6 7 8 9 10 11 12</td>
<td>4 5 6 7 8 9 10</td>
</tr>
<tr>
<td>9 10 11 12 13 14 15</td>
<td>13 14 15 16 17 18 19</td>
<td>11 12 13 14 15 16 17</td>
</tr>
<tr>
<td>16 17 18 19 20 21 22</td>
<td>20 21 22 23 24 25 26</td>
<td>18 19 20 21 22 23 24</td>
</tr>
<tr>
<td>23 24 25 26 27 28 29</td>
<td>27 28 29 30</td>
<td>25 26 27 28 29 30</td>
</tr>
<tr>
<td>30 31</td>
<td></td>
<td>31</td>
</tr>
</tbody>
</table>
1 Education: The Profile of the Bachelor in Data Science and Artificial Intelligence

1.1 What is the programme about

Knowledge is the central factor in modern society. Smart chips help companies to keep track of goods and to manage supplies and stocks. The usability of new high-tech communication electronics, such as smartphones, navigation devices and digital cameras, is greatly enhanced by intelligent software. Electronic banking and e-commerce heavily rely on real-time access to large databases and require highly safe protocols for information exchange. Medical and biological engineering helps medical doctors to arrive quickly at an accurate diagnosis, and is now opening up exciting new possibilities for personalized medicine. In the Data Science and Artificial Intelligence programme, you learn how to approach new challenges in these important areas. You learn to collect and organize valuable information, with the help of mathematics and modern computer techniques. You learn to use models and computers to analyse the information and to generate new knowledge, to draw important conclusions, to solve practical problems efficiently, and to speed up the decision-making process.

The Data Science and Artificial Intelligence programme offers you a unique combination of courses in applied mathematics, computer science and artificial intelligence. From the computer science point of view, the emphasis is on software, programming, algorithms and logic. Courses in applied mathematics are designed so that you quickly become acquainted with the important concepts, methods and techniques - always from a practical point of view, as proofs are never taught for mathematics sake only. Artificial intelligence provides you with ways to reason with available knowledge, and it introduces you into the world of machine learning and intelligent search. Did you know that, nowadays, artificial intelligence is a standard component of computer games?

What makes the course programme stand out from traditional educational programmes in computer science or mathematics, is the goal-oriented, practical and applied character. Mathematical and computer models continuously serve this goal, by bridging the gap between theory and practice. Classic optimization methods and modern intelligent techniques allow you to arrive at efficient and sometimes surprisingly elegant solutions. They help you to value the methods and techniques you encounter, and to gain insight into their usefulness for dealing with practical problems.

This unique profile of Data Science and Artificial Intelligence is the key to a flexible educational programme that prepares you for a successful professional career in modern society. The wide range of application areas and the project-centred educational model, offers you the opportunity to experience a diversity of topics from the ICT and telecommunications industry, medicine and biology, science, economics and business management, just to name a few. Optional courses, your choice of topic for the bachelor’s thesis, internships, and the possibility to study abroad, allow you to tune the contents of your study to your personal interests.

1.2 Study System

1.2.1 Project Centered Learning

The programmes of the Department of Data Science and Knowledge Engineering are designed around the Project-Centred Learning (PCL) teaching method. It resembles the Problem Based Learning style, for which Maastricht is well known. The PCL educational model is small-scale and student-oriented. You work in small groups on complex and challenging semester projects that require you to develop a variety of skills. Companies and institutes – who often submit projects - give our students the opportunity to gain invaluable experience by applying
their education to finding solutions to real-world problems. Therefore, you immediately apply what you have learned from the course material and lectures to real-life problems. Together with fellow students, you research which information is required and how it is best presented. At the end of each project, you deliver a functional computer programme and present your findings to your fellow students, the teachers and/or the client.

Project-Centred Learning has advantages:
• from the beginning you find out what teamwork means
• you learn project-related skills in a natural way
• you will be continuously placed in an active role
• you will be able to match theory with its applications
• PCL increases the student’s motivation

1.3 Study Abroad

The Department of Data Science and Knowledge Engineering has one of the highest ratios of international students. More than 75% of the scientific staff and 70% of the students are non-Dutch, giving rise to an international study environment. Additionally, DKE hosts a number of international exchange students each year and offers its own students a number of opportunities for international experiences themselves. For example, DKE offers its students the opportunity to study abroad for a semester during the elective semester. To this purpose, we collaborate with well-established partner universities.

For bachelor students there is an option to participate in an international study abroad programme during the fifth (elective) semester of the bachelor’s programme. For master students there is an option to participate during the third (elective) semester of the master’s programmes.

Please note that for students entering the Master programmes during the February intake, fewer universities may be available. Additional international collaboration is done through the exchange of course coordinators with RWTH Aachen University, with whom we closely collaborate. Furthermore, our two Master programmes Artificial Intelligence and Data Science for Decision Making are embedded in the School of Information Technology (SIT) of the transnational University Limburg (tUL).

More documentation and information about the participating exchange partner universities is available on the student portal under the “Going Abroad” section (https://intranet.maastrichtuniversity.nl/en/dke-students/going-abroad).

For additional information, you can contact the international relations officer or your study advisor at dke-studyadvice@maastrichtuniversity.nl

1.4 Degree

A successful conclusion of the bachelor’s programme will provide you with a bachelor’s certificate according to Dutch law, that is a ‘Bachelor of Science’. A successful conclusion of a master’s programme will provide you with a master’s certificate according to Dutch law, that is a ‘Master of Science’.

1.5 Study Feasibility and Quality Assurance

DKE strives for continuous improvement of the quality and feasibility of its study programmes. Student evaluations of each course help us maintaining a high standard of educational quality and keeping the study programmes feasible. The study programme’s feasibility means that a student with an appropriate background should be able to finish the study within the set number of years.

To maintain this standard, the quality assurance officer collects information about teaching, learning and assessment at the end of each period. The quality assurance officer then reports the outcomes of the student evaluations to the DKE Education Programme Committee (EPC), which also includes four student-representatives (two bachelor and two master students, one for each master programme). If the outcomes are unsatisfactory, the EPC will take action to improve the quality of a specific course (or project) or of the study as a whole. Therefore, student responses are essential in pointing out strong aspects and aspects for improvement of all educational activities.
As a DKE student, you are encouraged to give your opinion about each course, as it may help improve that course. Moreover, future students may benefit from the results and comments of your evaluation just as you may benefit from course evaluations of fellow students.

1.6 Courses at other faculties or universities

Bachelor and Master

If a student from DKE would like to participate in courses at other faculties or other universities, approval from the Board of Examiners is needed in advance. For more information regarding the Bachelor and Masters ask the study advisors: dke-studyadvice@maastrichtuniversity.nl

The Educational Minor leads to a limited second-degree teaching qualification. If you have successfully finished the minor, you are qualified to teach at lower secondary schools of VWO, HAVO and VMBO-tl (MAVO) level. Students in the BSc Data Science and Artificial Intelligence can - upon successful completion - acquire a teaching qualification for the main subject of Mathematics. The Educational Minor will be organised in close cooperation with the Fontys Leraren Opleiding (Teacher Training) in Sittard (FLOS) and Tilburg (FLOT). The main language of the Educational Minor is Dutch.

The programme contains several pedagogical-didactic courses in semester 5, along with education aimed at teaching methodology. There is also a mandatory practical internship, in the form of work placements, which is spread out over semester 5 and 6. The education meetings mostly take place at UM and occasionally at the Fontys Leraren Opleiding in Sittard. The practical internship will be done at several secondary schools in the whole province of Limburg and will continue until the end of semester 6. During the practical internship, the student spends one day a week at a secondary school for the course of a full school year. In this way, the necessary teaching experience is obtained. Knowledge and practice are closely connected in the Educational Minor.

Successful completion of the educational minor yields 35 ECTS of which five are extracurricular. This means that these 5 ECTS cannot be used to replace any other components of the original bachelor program. The minor is UM wide and is also open to students of other UM faculties. Students in the BSc Data Science and Artificial Intelligence who wish to participate in the program should have accumulated, by the end of their second year, all 60 ECTS from the first-year components and at least 52 ECTS from the second year components. Prior to their enrolment in this minor, a motivated request for participation has to be submitted in Dutch to the Board of Examiners (dke-boe@maastrichtuniversity.nl). Enrolment is dependent on selection and prior permission of the Board of Examiners.

If you have any questions regarding the contents of this educational minor, then please contact Prof. Dr. Frank Thuijsman, (f.thuijsman@maastrichtuniversity.nl).
# 2 Bachelor

## 2.1 Curriculum of the First Year of the Bachelor’s Programme

<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Computer Science 1</td>
<td>Introduction to Computer Science 2</td>
<td>Data Structures and Algorithms</td>
<td>Logic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete Mathematics</td>
<td>Calculus</td>
<td>Linear Algebra</td>
<td>Numerical Mathematics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to Data Science and Artificial Intelligence</td>
<td>Computational and Cognitive Neuroscience</td>
<td>ICT and Knowledge Management</td>
<td>Software Engineering</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PROJECT**  

## 2.2 Curriculum of the Second Year of the Bachelor’s Programme

<table>
<thead>
<tr>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases</td>
<td>Machine Learning</td>
<td>Theoretical Computer Science</td>
<td>Philosophy and Artificial Intelligence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probability and Statistics</td>
<td>Linear Programming</td>
<td>Mathematical Modelling</td>
<td>Simulation and Statistical Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Graph Theory | Reasoning Techniques | Human Computer Interaction and Affective Computing | 1 out of the following two electives:  
  * Natural Language Processing  
  * Introduction to Image and Video Processing |  |

**PROJECT**  

## 2.3 Curriculum of the Third Year of the Bachelor’s Programme

<table>
<thead>
<tr>
<th>Period 1*</th>
<th>Period 2*</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5 &amp; Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic Web</td>
<td>Large Scale IT and Cloud Computing; Logic for AI; Recommender Systems</td>
<td>Data Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game Theory</td>
<td>Introduction to Bio-Informatics</td>
<td>Operations Research Case Studies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robotics and Embedded Systems; Digital Society</td>
<td>Software and Systems Verification</td>
<td>Intelligent Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel Programming</td>
<td>Quantum Computing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PROJECT**  

**BACHELOR’S THESIS**
Third year students choose 6 optional courses (3 per period) in addition to the semester project in semester 1 of year 3. Students can also choose (1) elective courses at other UM bachelor programmes for at most 18 ECTS (2) the minor Entrepreneurship or (3) the educational minor. (4) In addition students can study abroad for a semester at one of our exchange partners (see internationalization section below). Please contact the study advisor for more information. Also, check the Study Abroad section in the “My Organisations” section of the Student Portal.

Periods 1 and 2 last eight weeks each in total. During week 1-7 there are classes and in week 8 exams. Periods 4 and 5 will last 9 weeks due to several holidays in this period. Three courses are offered during each period, each course is good for 4 credits (ECTS). Per course, five to seven hours of class are offered each week in year 1, and five hours in year 2 and 3. Note that year 1 has an attendance requirement: check the Education and Examination Regulations (EERs) that are published in this study guide and in the “My Organisations” section of the Student Portal for more information.

Next to these courses, you participate in a group project of 6 credits that will last the whole semester. Skill trainings and project meetings are mandatory: you are expected to be present during 100% of the skill classes and 100% of the project meetings in each academic year. If a student fails to do so, the project examiners may deviate from the group grade for this individual student. The project of semester 1 runs during period 1, 2 and 3. The project of semester 2 runs during period 4, 5 and 6. For specific details on the project curriculum in year 1 see section Project 1-1 and Project 1-2 with the course descriptions of year 1.

Periods 3 and 6 last three weeks, during which students work fulltime to finish their project assignment. After the project weeks, it’s possible to resit the failed courses from the previous two periods during the fourth week.

This project assignment is announced in the beginning of periods 1 or 4, along with the group composition. At the end of week 7 of the 8-weeks periods, each group separately gives a brief presentation for the examiners of the project. Each group shows their interpretation of the problem and will discuss their approach and schedule for the following phase. The examiners give their feedback to the content and progress of a project. These interim presentations will be assessed. The project is concluded at the end of the project weeks with the handing in of the final report, handing in of the product, and with an oral presentation of the project’s results for all groups. The assessment of the report, the presentation, and the product in principle result in the same mark for all the group members.

The final stage of your bachelor’s programme, period 5 and 6 of year 3 is reserved for writing your bachelor’s thesis that equals 18 credits. Every student has to conduct a short scientific research focussed on a relevant topic. This can be empirical or theoretical research. Students have acquired information on these different research domains throughout their educational programme. Each student has to hand in a signed bachelor thesis project plan to the Bachelor’s thesis coordinator. Each student is supervised by a thesis supervisor. In the second period of this semester, the students conduct their own research. At the bachelor thesis winter or summer conference, students have to present their results.
2.1 Curriculum of the First Year of the Bachelor’s Programme

In order to learn how the processing and preparation of knowledge is done with the help of computer systems, a thorough basic knowledge of specific mathematics and computer science subjects is required. This means that the first year is largely filled with mathematics and computer science subjects. Apart from that, you will also get an introduction to Computational & Cognitive Neuroscience and ICT & Knowledge Management.

The year is divided into four periods of eight weeks with three courses each, and two periods of four weeks during which you will work on a project. Each project is preceded by partial project assignments during the other periods. The week schedule works with two-hour clusters. In the overview below, the courses are indicated, as well as the study load in credits (ECTS). One ECTS stands for about 28 hours of study time (lectures, meetings and self-study). Besides the lectures that are given on the subjects, there will also be practicals and skills training.

<table>
<thead>
<tr>
<th>Year 1</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Period 1.1</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction to Data Science and Artificial Intelligence (KEN1110)</td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Computer Science 1 (KEN1120)</td>
<td>4</td>
</tr>
<tr>
<td>Discrete Mathematics (KEN1130)</td>
<td>4</td>
</tr>
<tr>
<td>Project 1-1 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 1.2</strong></td>
<td></td>
</tr>
<tr>
<td>Computational &amp; Cognitive Neuroscience (KEN1210)</td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Computer Science 2 (KEN1220)</td>
<td>4</td>
</tr>
<tr>
<td>Calculus (KEN1440)</td>
<td>4</td>
</tr>
<tr>
<td>Project 1-1 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 1.3</strong></td>
<td></td>
</tr>
<tr>
<td>Project 1-1 (KEN1300)</td>
<td>6</td>
</tr>
<tr>
<td><strong>Period 1.4</strong></td>
<td></td>
</tr>
<tr>
<td>Linear Algebra (KEN1410)</td>
<td>4</td>
</tr>
<tr>
<td>Data Structures and Algorithms (KEN1420)</td>
<td>4</td>
</tr>
<tr>
<td>ICT &amp; Knowledge Management (KEN1430)</td>
<td>4</td>
</tr>
<tr>
<td>Project 1-2 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 1.5</strong></td>
<td></td>
</tr>
<tr>
<td>Numerical Mathematics (KEN1540)</td>
<td>4</td>
</tr>
<tr>
<td>Software Engineering (KEN1520)</td>
<td>4</td>
</tr>
<tr>
<td>Logic (KEN1530)</td>
<td>4</td>
</tr>
<tr>
<td>Project 1-2 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 1.6</strong></td>
<td></td>
</tr>
<tr>
<td>Project 1-2 (KEN1600)</td>
<td>6</td>
</tr>
</tbody>
</table>

(*) Project 1-1 will start in period 1.1 and will run until period 1.3; Project 1-2 will start in period 1.4 and will run until period 1.6. The credits for the projects will become available at the end of period 1.3 and period 1.6, respectively. Please see the course description section Project 1-1 and Project 1-2 for more details on the project curriculum. For each period, we will give a short explanation of the various parts. Before the start of each period, the students will receive detailed information about the content, the study material, the teaching form, the schedule, and the examination method.
Introduction to Data Science and Artificial Intelligence (Code: KEN1110)

Examiner: Dr. R. Cavill, Dr. P. Bonizzi and Prof. Dr. A. Wilbik

Desired Prior Knowledge: None.

The course appears as desired prior knowledge for the courses Reasoning Techniques and Theoretical Computer Science.

Prerequisites: None.

Course description: The course Introduction to Data Science and Artificial Intelligence offers a comprehensive overview of the core topics in Data Science and Artificial Intelligence (DKE), both from a mathematical and from a computational perspective. Particular emphasis is put on the basic classes of techniques and methods, the theoretical underpinnings of data science and computational intelligence, and some example application domains of data science. As such, the course provides an overview of many topics that are addressed in much more detail throughout the Bachelor's programme.

Knowledge and understanding: After successful completion of the course, students will be able to recognise what real world problems require the use of data science, and approach their solution by using a data science process, namely: explore the data, model the data, and perform simulations if required. Moreover, they will exhibit knowledge in the basic concepts of artificial intelligence, such as agents, search, artificial intelligence, decision trees.

Applying knowledge and understanding: Students learn to recognise applications of data science and artificial intelligence in different domains and apply the basic techniques they have learnt from both.

Making judgements: Upon completion of the course, students are able to recognise the relevant domains of data science and artificial intelligence when confronted with data science and artificial intelligence problems.

Communication: Students are able to explain the process they used to generate results and communicate the meaning of those results in context.

Learning skills: Students will be able to recognise small-scale data science problems and autonomously and critically reflect upon the appropriateness of the data science process for tackling those, and propose a primary solution.

Study material: Material will be provided during the course.

Recommended literature:

Exam: There will be a closed book written exam at the end of the course.

ECTS: 4

Introduction to Computer Science 1 (Code: KEN1120)

Examiner: Dr. G. Spanakis and dr. E. Hortal Quesada

Desired Prior Knowledge: None. The course appears as desired prior knowledge for the courses Introduction to Computer Science 2, Data Structures and Algorithms, Software Engineering, Databases and Machine Learning.

Prerequisites: None. It appears as part of the pre-requisites of the second semester project in year 1, both projects of year 2 and the third year courses Prolog, Parallel Programming and Robotics.

Description: The course provides the basics of computer science and computer programming. After a short introduction to computer organization, the principles of programming are presented. The main topics of the course are: data types, variables, methods, parameters, decision structures, iteration, arrays, recursion and a branching application (related to the semester project). Programming skills will be acquired during practical sessions using the object-oriented programming language Java.

Knowledge and understanding: The course offers preliminary methodological and theoretical bases for studying and applying computers and computer programming on which the rest of the curriculum builds.
Applying knowledge and understanding: Whenever a computer system or a programming system has to be designed and implemented the knowledge and insights acquired during the course can be used and applied.

Making judgements: After successful completion of the course, students will be able to judge the quality and correctness of simple non-object-oriented programs.

Communication: The skills acquired during the course will enable student to communicate about standard programming constructs and algorithmic basics.

Learning Skills: After successful completion of the course students will be able to formalize, analyse and program solutions to simple software problems.


Exam: Closed-book written exam (80%) + Assignments (20%)

ECTS: 4
Computational and Cognitive Neuroscience (Code: KEN1210)

**Examiner:** Dr. A. Roebroeck and Dr. M. Capalbo

**Desired Prior Knowledge:** None.

**Prerequisites:** None.

**Description:** The course Computational and Cognitive Neuroscience presents an overview of the core topics in cognitive and biological psychology. These topics include (human) perception, learning, memory, planning, problem solving, reasoning, language, speech, and action. Both the functional and neuroanatomical foundations of cognitive faculties are addressed. Several models of cognition and theories of brain function that are of relevance to knowledge engineering will be outlined. Several skills trainings will be given to train understanding in biological functioning of neuronal communication, and functioning of neural networks and genetic algorithms.

**Knowledge and understanding:**

The student can recount the main points of the domain of cognitive science

- The student can describe the main points of the domain of cognitive science
- The student can explain the following (human) behaviours while using these points: perception, learning, memory, planning, problem solving, reasoning, language, speech, and action
- The student can identify the computational aspects and computational applications of these fields

**Applying knowledge and understanding:**

- This knowledge is applied in in two practical assignments in which the students are asked to create a genetic algorithm and a neural network

**Making judgement:**

- Upon completion of the course, students are able to interpret data and literature about a subject in (or related to) the domain of cognitive and biological psychology.
- Using the data and literature, they can support judgements about the societal, scientific or ethical aspects of the subject.

**Communication:**

- Students are able to communicate ideas and solutions to an audience of non-experts and experts.

**Learning skills:**

- Students have acquired the skill to translate theoretical models into computational models.

**Study material:** Material will be provided during the course.


**Exam:** Written exam

**ECTS:** 4

Introduction to Computer Science 2 (Code: KEN1220)

**Examiner:** Dr. E. N. Smirnov and Dr. E. Hortal Quesada

**Desired prior knowledge:** Basic Java Programming

**Prerequisites:** None.

**Description:** This course is a follow-up of the course Introduction to Computer Science 1. It teaches object-oriented programming in Java. The main topics covered in the course are objects and classes, interfaces and polymorphism, event handling, inheritance, graphic user interfaces, exception handling, and streams.

Knowledge and understanding: After successful completion of the course, students will be able to explain the methodological and theoretical principles of object-oriented programming.
Calculus (Code: KEN1440)
Examiner: Dr. A. Briassouli and Dr. O. D’Huys
Prerequisites: None.
Description: The following subjects will be discussed in Calculus: limits and continuity, differential calculus, inverse and transcendental functions, mean value theorem, integral calculus, sequences and series, introduction to differential equations, introduction to multivariable calculus. In addition to the main facts and concepts, problem-solving strategies will be discussed. Both the intuition behind the concepts and their rigorous definitions will be presented along with simple examples of formal mathematical proofs. Knowledge and understanding: Student can define, write and explain key facts and concepts involving limits and continuity, can interpret and solve differential calculus, inverse and transcendental functions, mean value theorem, integral calculus, sequences and series, first-order linear differential equations, basics of multivariable calculus.
Applying knowledge and understanding: Students are able to provide examples of instances of problems of specific properties. Students are able to solve problems involving concepts learned in the course, using standard problem-solving strategies. Students are able to understand simple mathematical proofs.
Making judgements: Students are able to analyse a simple problem within the course content and justify the solution methodology they choose. They can summarize this methodology mathematically.
Communication: Students are able to explain their solution strategy in written form and defend their solution strategy in discussion with others.
Learning Skills: After successful completion of the course the students will be able both to solve standard problems (constructing graphs of functions, finding extrema of functions, computing limits, summing infinite series etc.) and to apply their knowledge in solving and analysing more complex problems (e.g. in analysis of numerical algorithms).
Study material: Calculus, a complete course, any edition, by R.A. Adams, Addison Wesley Longman and materials provided during the lecture.
Exam: One intermediate online assessment and a final written exam.
ECTS: 4
Project 1.1

Course title: Project 1-1 (KEN 1300)

Examiner: Dr. S. Chaplick, Dr. E. Smirnov, Dr. E. Hortal Queasada, Dr. M. Staudigl, Dr. J. Niehues and Dr. T. Rienstra

Coordinator: Dr. K. Schneider

Tutors: Dr. K. Schneider, Dr. M. Popa

Description: Students work on a project assignment in small groups of about six students. The group composition stays the same for the whole project and is announced shortly before the project opening in period 1.1. The students are guided through the project by a fixed tutor. The project assignment is divided into three subtasks (one per period) and is strongly related to the content of the courses from period 1.1 and 1.2. In period 1.1, after receiving the assignment for the whole project at the end of week 5, the students work full-time on the project in week 6. In this week, each group meets the tutor twice. In period 1.2, the students continue working on the project, while also having to attend the courses of that period. They meet their tutor approximately once a week. In period 1.3, the students work three weeks full-time on the project and meet their tutor about twice a week.

At the beginning of period 1.2 and 1.3, the students have to hand in a planning for the current phase. At the end of each period, the students have to give a presentation and the source code, presentation and an overview of who did what need to be uploaded to Canvas. While the presentations at the end of period 1.1 and 1.2 are in front of the examiners and the tutors, the presentations at the end of period 1.3 will additionally be in front of the fellow students. In period 1.3, they furthermore have to hand in a report and attend a product and report examination.

Knowledge and understanding: Interpret the meaning of mathematical models of real-world processes. Gain insight into practical use of software design and development principles. Recognise and relate user-computer interactions to concepts from graphics and user-interface frameworks. Strengthen knowledge of basic algorithms and methods for specific problems in artificial intelligence and applied mathematics.

Applying knowledge and understanding: Students will be able to design an answer strategy for scientific questions using analytical thinking and logical reasoning and to translate mathematical models to software code. Furthermore, students will be able to implement software to solve problems in applied mathematics by applying numerical methods and artificial intelligence algorithms, formulate computational experiments, and analyse and interpret the results, apply design and development principles in the construction of software systems and use existing software application frameworks for graphics and user interfaces. Even more so, students will learn to use tools for software project management such as version control systems and issue trackers, identify project goals, deliverables, and constraints. Lastly they will learn how to plan and chair meetings, create notes for minutes, work in a team such that the workload is balanced and plan teamwork by setting deadlines and distributing tasks.

Making judgements: Students will learn to evaluate different mathematical and computational models with respect to their suitability, efficiency and correctness for a specific task.

Communication: Students will be able to give a clear and well-constructed presentation, including a demonstration of the product, and with appropriate use of illustrations and/or videos, to offer and respond to questions on and constructive criticism of presentations. Furthermore, they will learn to write a project report according to the structure of an academic article, submit arguments in exact sciences, with appropriate use of formulae and figures. They learn to cite published sources in the project report according to the academic guidelines. Additionally, students will learn to structurally inform stakeholders on project progress and effectively communicate with project group members about task division, planning and project deadlines, effectively communicate with group members by listening to others’ ideas; be contactable include others in the discussion. It will be important to cooperate in a group to reach a consensus view, communicate in the English language, elicit and evaluate relevant scientific background information.
Learning skills: Reflect on one's own academic abilities and functioning in a team.

Study material: Project manual project 1-1, Maastricht University, DKE

Assessment: Final grade = 0.15 x grade of phase 1 + 0.15 x grade of phase 2 + 0.7 x grade of phase 3, where grade of phase 3 = 40% grade for product + 40% grade for report + 10% grade presentation + 10% grade project management. Missing project meetings and skill classes will cause an individual grade reduction of the final grade. Furthermore, the examiners may deviate from the group grade for individual students if they either contributed significantly more or less than other group members.

Skill classes:

Introduction to Project Work (period 1.1)
The students learn how DKE applies the PCL approach. The structure of the periods regarding the project is explained. Furthermore, the students will get to know their team members and will be introduced to various aspects of project work. Furthermore, the students will work on a small assignment in groups and reflect on their group work.

Elementary writing and presenting (period 1.2)
This skill class is divided into two parts: 1) Elementary Writing: Here, you learn how to structure a report/article and what content to write in each of the sections of a report. You learn how to use signal words, referral words and basics of how to write in the mathematical language. 2) Presenting: In this part, you learn basic knowledge about body language, group presentations and slide layout.

Citing and Referencing (period 1.2)
This skill class provides you with basic knowledge about when and how to cite.

Team Dynamics “How to effectively communicate and cooperate in my team?” (period 1.3)
In this skill class, you will learn effective communication strategies that will help you to cooperate in your team.

Advanced Presenting (period 1.3)
In this skill class, you will present within your project group and receive individual feedback on presentation style and content.

ECTS: 6
Linear Algebra (Code: KEN1410)

**Examiner:** Dr. S. Chaplick and Dr. M. Musegaas

**Tutor(s):** None.

**Desired Prior Knowledge:** None. The course itself occurs as part of the pre-requisites of the second semester project in year 1, and as desired prior knowledge for the second year courses, Mathematical Modelling, Linear Programming.

**Prerequisites:** None.

**Course description:** This course introduces the fundamental concepts of linear algebra, and examines them from both an algebraic and a geometric point of view. First, we address what can be recognized without doubt as the most frequently occurring mathematical problem in practical applications: how to solve a system of linear equations. Then we discuss linear functions and mappings, which can be studied naturally from a geometric point of view. Vectors spaces are then introduced as a common framework that brings all themes together. Next, we shift from the geometric point of view to the dynamic perspective, where the focus is on the effects of iterations (i.e., the repeated application of a linear mapping). This involves a basic theory of eigenvalues and eigenvectors, which have many applications in various branches of science as for instance in problems involving dynamics and stability, in control theory, and in optimization problems found in data science. Key concepts in the course are vectors, matrices, systems of linear equations, eigenvalues, eigenvectors, linear transformations, and orthogonality. The software package Matlab is introduced in the accompanying computer classes, where emphasis is put on the application of linear algebra to solve real world problems.

**Knowledge and understanding:** Students are able to recognize and explain the fundamental concepts of Linear Algebra: systems of linear equations, vectors and vector spaces, basis and coordinates, matrices and matrix-vector computations, linearity and orthogonality, linear independence, rank, fundamental spaces (row space, column space, and null space), determinants and invertibility, eigenvalues and Eigen spaces, diagonalization.

**Applying knowledge and understanding:** Students are able to analyse a linear algebra problem from both an algebraic and a geometrical point of view. Students can solve systems of linear equations, compute determinants and rank, compute eigenvalues and Eigen spaces, make use of complex numbers, diagonalizable matrices, and perform change of coordinates.

**Making judgements:** Students are able to look at the same problem from different angles and to switch their point of view (from geometric to algebraic and vice versa).

**Communication:** Students are able to motivate both from an algebraic and a geometric point of view the solution set of a system of linear equations, the linear independence and orthogonality of a set of vectors, the linear transformation between two coordinate systems, the fundamental spaces associated with a matrix, the invertibility of a matrix, and the diagonalization of a matrix in terms of the properties of its eigenvalues and eigenvectors.

**Learning skills:** Students have acquired the skills to autonomously recognize elements of practical problems, which can be addressed and solved with linear algebra, and use Matlab to solve larger scale problems.


Recommended literature: None.

**Exam:** There will be a closed book written exam at the end of the course.

**ECTS:** 4
Data Structures and Algorithms (Code: KEN1420)

Examiner: Dr. J. Niehues and T. Pepels, M.Sc.

Tutor(s): TBA.

Desired Prior Knowledge: Discrete Mathematics, Introduction to Computer Science 1 and 2. The course is desired prior knowledge for Theoretical Computer Science.

Prerequisites: None. The course itself occurs as part of the pre-requisites of both projects of year 2 and the third year course Parallel Programming.

Description: As a continuation of the courses Computer Science 1 and 2, this course will treat the systematic design and application of data structures and algorithms. Data structures such as lists, trees, graphs, and strings, the associated algorithms and their complexity will be treated. Design principles for algorithms such as recursion, divide-and-conquer and dynamic programming will be treated as well.

Knowledge and understanding: Students are able to give examples of data structures and explain do they support program design. Students are able to name what types of standard data structures exist and illustrate their properties. Students are able to describe some standard algorithms and highlight their properties. Students are able to illustrate how to develop and analyse new algorithms.

Applying knowledge and understanding: Students are able to select the appropriate data structure for a given problem and students are able to propose an algorithm for solving a given problem.

Making Judgements: Students are able to justify if and determine how data structures should be applied. Furthermore, students are able to assess whether algorithms are appropriate and efficient.

Communication: Students are able to explain how data structures and algorithms are to be included in program designs.

Learning Skills: Students are able to reflect on which data structures and/or algorithms are applicable for each problem.


Exam: ‘Closed Book’ written exam, during the course the students will receive a number of assignments, which can earn them up to a total of one bonus point.

ECTS: 4
ICT and Knowledge Management (KEN1430)

**Examiner:** Dr. Ir. Peter Bollen (SBE)
**Tutor:** Dr. Ir. Peter Bollen (SBE)

**Description:** Knowledge is a fundamental prerequisite in the ability of a person to execute a task. This ability consists of explicit knowledge or information, implicit knowledge or experiences, skills and attitudes.

In this course, we will focus on the (fact-based) conceptual modelling approach in which we also clearly make a link to a relevant contemporary domain in computer science: block chain. We will use the following definition of Knowledge Management:

Knowledge management is an integral approach for the identification, the structuring, the sharing and evaluation of knowledge in the organization.

**Knowledge and understanding:** Students will be able to explain fundamental building blocks of a domain knowledge model. Students can describe the development of the semantic-conceptual modelling approach to knowledge management from the 1970s until now. Students can describe and outline the semantic-conceptual knowledge modelling process.

Applying knowledge and understanding: Students will be able to apply the steps in the semantic-conceptual knowledge modelling process on scaled-down examples. Students will be able to apply all steps in the semantic-conceptual knowledge modelling process on ‘real-life’ modelling domains.

**Making Judgement:** Students will be able to assess a knowledge domain and determine whether semantic-conceptual modelling can be applied and what domain knowledge and domain examples will be needed to derive the fact types and integrity rules.

**Communication:** Students will be able to write a professional report in which they communicate their findings to a domain problem that they have solved by understanding, assessing and applying the semantic conceptual modelling knowledge they have acquired in (the first part of) this course.

**Learning Skills:** Students will be able to apply a professional way of working according to a well-documented methodology.

**Study material:** E-reader

**Exam:** hand-in exercises, mid-term open book exam, final group project

**ECTS:** 4
**Period 1.5**

**Numerical Mathematics (Code KEN1540)**

**Examiner:** Dr. P. Collins and Dr. M. Boussé

**Tutor(s):** Dr. P. Collins

**Desired prior knowledge:** calculus, linear algebra

**Description:** Numerical mathematics is the art of solving mathematical problems with the aid of a digital computer. In this course, we will cover the fundamental concepts of numerical mathematics, including the floating-point representation of real numbers, truncation and round off errors, iterative methods and convergence. We will study the simplest and most important algorithms for core problems of numerical mathematics, namely the solution of algebraic equations, interpolating data by polynomials and splines, numerically estimating derivatives and integrals, solving differential equations, approximating functions by polynomials and Fourier series, solving systems of linear algebraic equations and computing eigenvalues. There will be a strong practical component, with students being expected to write their own numerical code and test the performance and suitability of different methods on various problems.

**Knowledge and understanding:** By the end of this course, students will have knowledge of the fundamental problems of numerical mathematics and basic techniques for their solution. You will understand issues of efficiency and numerical accuracy, will be able to analyse which numerical methods are likely to perform best on different types of problem, and evaluate whether the results of a given computation are trustworthy. You will be able to write your own code (in MATLAB) implementing basic numerical algorithms. Advanced students will have the skills necessary to adapt existing numerical algorithms and develop new algorithms.

**Applying knowledge and understanding:** Students will be expected to implement the algorithms covered in the lectures, and apply these to practical problems.

**Making judgements:** Students will learn how to analyse which numerical methods are likely to perform best on different types of problem, and to evaluate whether the results of a given computation are trustworthy.

**Communication:** Students will learn the terminology required to discuss numerical algorithms and the results of numerical computations with mathematicians, (social) scientists and engineers.

**Learning skills:** Students will learn to design, analyse, implement and apply numerical methods.

**Study material:** Slides, exercise sheets.

**Recommended literature:** Faires & Burden, “Numerical Methods”.

**Exam:** Written examination with formula sheet (80%). Computer-based homework exercises (20%+10% Bonus).

**ECTS:** 4
Software Engineering (Code: KEN1520)

Examiner: Dr. C. Seiler, T. Pepels, M.Sc.

Desired prior knowledge: Introduction to Computer Science 1 and 2, Data Structures and Algorithms.

Prerequisites: None

Description: This course introduces students to software design and project management concepts. Students are introduced to multiple techniques they require to work on medium and large-scale projects in professional business and research environments. Students learn how to produce professional, reliable, and cost-efficient software that can be developed in a team, reused, maintained, further evolved, and that is tested professionally. Covered concepts include requirement engineering, project planning, risk management, software evaluation and testing, software engineering processes, design principles, software architectures, design patterns, code review, version control, specifications, debugging, and abstract data types.

Knowledge and understanding: Students learn how to efficiently design and write professional software that meets specifications made by themselves or by customers. Students learn what the essential elements of the software engineering process are. These elements include requirement analysis, design methodologies, implementation strategies, and validation techniques.

Applying knowledge and understanding: Students acquire the skills

• to critically analyse software requirements, software designs, software implementations, and software evaluations.
• to efficiently plan, execute, and monitor progress in-group projects.
• to cooperate better in a group and to participate more effectively as a professional in academia or business environment.

Software engineering is a core activity of knowledge engineers and data scientists. In their professional career DKE students often will act as active programmers, software designers, and project managers where they need to create software as part of a team or act as team leader. The project management skills and tools being taught in this course that help students creating professional and cost-effective software are thus crucial for their career and lay the foundation for further studies of in this field. The knowledge obtained will directly help students for the subsequent semester projects and Bachelor thesis.

Making judgements: Students learn to judge the viability of selected software development methodologies and new developments in design concepts during their career. Students learn to compare design choices and judge their consequences.

Communication: Students learn to discuss and document software developments professionally. Knowledge about widely spread standard software development techniques and about standard design patterns are essential for efficient communication between software developers. Standards in software engineering facilitate cost-effective communication and help to avoid misunderstandings between customers and suppliers, between team leaders and team members as well as between team members.

Learning skills: Students learn to successfully reflect on their project management skills, on how they contribute to a software project as part of a team, and on how to adjust their software engineering approaches to different professional scenarios. Students learn to reflect on and to verify own and others software designs and implementations in a professional manner.

Study material: Lecture material provided during the lecture.

Recommended literature:

• Gamma et al., Design Patterns: Elements of Reusable Object-Oriented Software (1994)

Exam: Written “closed-book” exam at the end of the course. During the course, students receive several graded assignments that can earn them a maximum bonus grade of 1.0.

ECTS: 4
Logic (Code: KEN1530)

Examiner: Dr. Ir. N. Roos and Dr. T. Rienstra

Prerequisites: None. The course appears as a prerequisite for the course Logic for AI.

Description: This course deals with three logical systems, namely propositional logic, first-order predicate logic and epistemic logic. The course covers notation systems, syntax and semantics, valid consequences, deduction, semantic tableaux, and proof systems.

Knowledge and understanding: Students need to get accustomed to the fundamental concepts of mathematical logical systems (propositional logic and predicate logic) to able to describe information in a logical framework and to reason and prove correctly. Students will get accustomed to the basic concepts of some advanced logical systems (epistemic logic and dynamic logic).

Applying knowledge and understanding: Student will apply the reasoning and proof methods learned to small-scale problems and some more complex situations.

Making judgements: Students will learn to judge how to reason correctly using mathematical proofs and how to judge which logical system is suitable to solve the problem at hand.

Communication: The chosen syntax of the logical language used must be easily understandable by peers and others experts the logical proofs given must be correct, concise and easily understandable.

Learning skills: having learned basic logical concepts and reasoning techniques the students are able to apply them to larger-scale problems.

Study material:
- Johan van Benthem, Hans van Ditmarsch, Jan van Eijck, Jan Jaspars, Logic in Action. Edition of February 2014 or later. This is a freely available e-book. Check your Student Portal for the link.

Exam: Written exam; during the course the students will receive three assignments, that, if they receive a sufficient grade, may earn them up to a total of one bonus point.

ECTS: 4
Period 1.6

Course title: Project 1-2 (KEN 1600)

Examiners: Dr. P. Collins, Dr. N. Roos, Dr. C. Seiler
Coordinator: Dr. K. Schneider
Tutors: Dr. K. Schneider, Dr. O. D’Huys
Prerequisites: In order to participate in this project the student has to have passed two out of four courses from the set: Discrete Mathematics, Linear Algebra, Computer Science I and Computer Science II.

Description: Students work on a project assignment in small groups of about six students. The group composition stays the same for the whole project and is announced before the project opening in period 1.4. The students are guided through the project by a fixed tutor. The project assignment is divided into three subtasks (one per period) and is strongly related to the content of the courses from period 1.4 and 1.5. In period 1.4, after receiving the assignment for the whole project at the end of week 5, the students work full-time on the project in week 6. In this week, each group meets the tutor twice. In period 1.5, the students continue working on the project, while also having to attend the courses of that period. They meet their tutor approximately once a week. In period 1.6, the students work three weeks full-time on the project and meet their tutor twice a week.

At the beginning of period 1.5 and 1.6, the students have to hand in a planning for the current phase. At the end of each period, the students have to give a presentation and the source code, presentation and an overview of who did what need to be uploaded to Canvas. While the presentations at the end of period 1.4 and 1.5 are in front of the examiners and the tutors, the presentations at the end of period 1.6 will additionally be in front of the fellow students. In period 1.6, they furthermore have to hand in a report and attend a product and report examination.

Knowledge and understanding: Interpret the meaning of mathematical models of real-world processes. Gain insight into practical use of software design and development principles. Recognise and relate user-computer interactions to concepts from graphics and user-interface frameworks. Strengthen knowledge of basic algorithms and methods for specific problems in artificial intelligence and applied mathematics.

Applying knowledge and understanding: Students will be able to design an answer strategy for scientific questions using analytical thinking and logical reasoning and to translate mathematical models to software code. Furthermore, students will be able to implement software to solve problems in applied mathematics by applying numerical methods and artificial intelligence algorithms, formulate computational experiments, and analyse and interpret the results, apply design and development principles in the construction of software systems and use existing software application frameworks for graphics and user interfaces. Even more so, students will learn to use tools for software project management such as version control systems and issue trackers, identify project goals, deliverables, and constraints. Lastly they will learn how to plan and chair meetings, create notes for minutes, work in a team such that the workload is balanced and plan teamwork by setting deadlines and distributing tasks.

Making judgements: Students will learn to evaluate different mathematical and computational models with respect to their suitability, efficiency and correctness for a specific task.

Communication: Students will be able to give a clear and well-constructed presentation, including a demonstration of the product, and with appropriate use of illustrations and/or videos, to offer and respond to questions on and constructive criticism of presentations. Furthermore, they will learn to write a project report according to the structure of an academic article, submit arguments in exact sciences, with appropriate use of formulae and figures. They learn to cite published sources in the project report according to the academic guidelines. Additionally, students will learn to structurally inform stakeholders on project progress and effectively communicate with project group members about task division, planning and project deadlines, effectively communicate with group members by listening to others’ ideas; be contactable include others in the discussion. It will be important to cooperate in a group to reach a consensus view, communicate in the English language, elicit and
evaluate relevant scientific background information.

**Learning skills:** Reflect on one's own academic abilities and functioning in a team.

**Study material:** Project manual project 1-2, Maastricht University, DKE.

**Assessment:** Final grade = 0.15 x grade of phase 1 + 0.15 x grade of phase 2 + 0.7 x grade of phase 3, where grade of phase 3 = 40% grade for product + 40% grade for report + 10% grade presentation + 10% grade project management. Missing project meetings and skill classes will cause an individual grade reduction of the final grade. Furthermore, the examiners may deviate from the group grade for individual students if they either contributed significantly more or less than other group members.

**Skill classes:**
- **Information Research:** Reference Tool (period 1.4)
  This skill class will give you an introduction to reference tools that will help you to manage your references.
- **Team Dynamics “Laying the foundation of effective teamwork” (period 1.4)**
  This team dynamics workshop aims to provide you with a deeper awareness, insight and practice in effective team collaboration & co-creation. During this introduction workshop, you and your project team will draw up a team charter (contract) to initiate effective group collaboration in project 1-2.
- **Advanced Presenting (period 1.4)**
  This skill class will focus on presentation skills and techniques. This introduction helps with public speaking and prepares you for project report presentations. Areas of focus include: structure of a presentation, public speaking techniques and enunciating, language aspects to remember while planning a presentation, and the dos and don’ts expected by Maastricht University.
- **Academic Writing (period 1.5)**
  In the project skills meetings you will explore the key structure of your report, as well as key points of Academic Writing at Maastricht University. Areas of focus include: structure of paper; linguistic aspects of writing in English, presenting information logically and citation and reference procedures.
- **Team Dynamics “Evaluating and drawing lessons from the project teamwork” (period 1.6)**
  In this skill class, you are going to evaluate the team collaboration and communication during project 1-2 by means of interactive exercises.

**ECTS:** 6
## 2.2 Curriculum of the Second Year of the Bachelor’s Programme

<table>
<thead>
<tr>
<th>Year 2</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Period 2.1</strong></td>
<td></td>
</tr>
<tr>
<td>Databases (KEN2110)</td>
<td>4</td>
</tr>
<tr>
<td>Probability and Statistics (KEN2130)</td>
<td>4</td>
</tr>
<tr>
<td>Graph Theory (KEN2220)</td>
<td>4</td>
</tr>
<tr>
<td>Project 2-1 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 2.2</strong></td>
<td></td>
</tr>
<tr>
<td>Reasoning Techniques (KEN2230)</td>
<td>4</td>
</tr>
<tr>
<td>Machine Learning (KEN2240)</td>
<td>4</td>
</tr>
<tr>
<td>Linear Programming (KEN2520)</td>
<td>4</td>
</tr>
<tr>
<td>Project 2-1 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 2.3</strong></td>
<td></td>
</tr>
<tr>
<td>Project 2-1 (KEN2300)</td>
<td>6</td>
</tr>
<tr>
<td><strong>Period 2.4</strong></td>
<td></td>
</tr>
<tr>
<td>Mathematical Modelling (KEN2430)</td>
<td>4</td>
</tr>
<tr>
<td>Human Computer Interaction and Affective Computing (KEN2410)</td>
<td>4</td>
</tr>
<tr>
<td>Theoretical Computer Science (KEN2420)</td>
<td>4</td>
</tr>
<tr>
<td>Project 2-2 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 2.5</strong></td>
<td></td>
</tr>
<tr>
<td>Philosophy &amp; Artificial Intelligence (KEN2120)</td>
<td>4</td>
</tr>
<tr>
<td>Simulation and Statistical Analysis (KEN2530)</td>
<td>4</td>
</tr>
<tr>
<td>Natural Language Processing (KEN2570) (Elective)</td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Image and Video Processing (Code KEN3238) (Elective)</td>
<td>4</td>
</tr>
<tr>
<td>Project 2-2 (*)</td>
<td>4</td>
</tr>
<tr>
<td><strong>Period 2.6</strong></td>
<td></td>
</tr>
<tr>
<td>Project 2-2 (KEN2600)</td>
<td>6</td>
</tr>
</tbody>
</table>

(*) Project 2-1 will start in period 2.1 and will run until period 2.3 with weekly meetings; Project 2-2 will start in period 2.4 and will run until period 2.6 with weekly meetings. The credits for the projects will become available at the end of period 2.3 and 2.6 respectively. For each period, we will give a short explanation of the various parts. Before the start of each period, the students will receive detailed information about the content, the study material, the teaching form, the schedule, and the examination method.
Period 2.1

Databases (Code: KEN2110)

Examiner: T. Pepels, M.Sc.

Desired Prior Knowledge: Data Structures and Algorithms, Software Engineering.

Prerequisites: Introduction to Computer Science 1 and 2.

Description: This course covers the use of (relational) databases and data modelling with the goal of writing (distributed) data-intensive software applications. Specifically, students will learn to use the Structured Query Language (SQL) to manipulate data to develop data-models that are Atomic, Consistent, Isolated and Durable. Moreover, the course covers alternative (distributed) data-storage methods and object persistence techniques such as NoSQL. During the course, students will learn to use different database management systems and how to use them to build software.

Knowledge and understanding: Students will be able to describe the basic concepts of databases, explain the fundamental concepts of database management systems, query languages, data modelling and database programming.

Applying knowledge and understanding: Students will be able to explain the proper database design based on system requirements, indicate possibilities and limitations of database types. In addition, students will be able to combine software architectures to design and construct a database application.

Making judgements: Students will be able to analyze and justify a practical database problem, examine different approaches, and refine database models based on use cases. Moreover, they can make improvements to existing database designs, reflect on certain solutions of the databases design and implementation, and assess the correctness of the database model.

Communication: Students will be able to summarize the basic entities and relationships involved in persistent data, and communicate with developers, database managers and users on proper database design and interfacing.

Learning skills: Students will be able to identify and understand follow-up literature, beyond the teaching material of the course.

Study material:

Recommended literature:

Exam: Written exam (75%) + practical assignment (25%)

ECTS: 4
Probability and Statistics (Code: KEN2130)

Examiner: Dr. C. Seiler
Prerequisites: None
Description: In this course, we will review basic concepts in statistical inference (confidence intervals, parameter estimation, and hypothesis testing), and the two main philosophies to make statistical inferences about the world (frequentist and Bayesian) and their combination (empirical Bayes). We will then study computer-intensive methods that work without imposing unrealistic or unverifiable assumptions about the data generating mechanism (the bootstrap and Markov chain Monte Carlo). This will finally provide us with the foundations to study modern inference problems in machine learning and statistics (selective inference, causal inference, and predictive inference).
Knowledge and understanding: Knowing a wide range of modern statistical models and computational tools to draw inferences will provide the foundations for analyzing complex data in academia and industry.
Applying knowledge and understanding: Students will be able to build statistical models and choose computational tools to perform inference.
Making judgements: In this course, we will discuss one of the most important aspects of analyzing data: being skeptical of results and avoiding wishful thinking.
Communication: Students will present their results using literate programming and reproducible workflows.
Learning skills: Students will be able to understand, apply, and extend papers from statistics journals and machine learning conferences.
Study material: Lecture slides, selected chapters from textbooks, and papers.
Exam: 20% homework assignments and 80% written final exam.
ECTS: 4

Graph Theory (Code: KEN2220)

Examiner: Dr. M. Mihalák
Desired Prior Knowledge: Discrete Mathematics; Data Structures and Algorithms
Prerequisites: None
Description: A graph is simply a collection of points, some of which are joined by lines. This deceptively simple structure is one of the cornerstones of both theoretical and applied computer science. A great many problems that arise in the real world can be modeled as graph problems. Several classical examples include the problem of finding the shortest route between two cities, of maximizing flow in a network of pipelines, or of finding an optimal pairing between producers and consumers. In this course we will look at both the algorithmic/applied side of graph theory and its more abstract mathematical foundations, because the latter is often important for understanding the former. We will cover topics such as paths, tours, trees, matchings, flows and colorings.
Knowledge and understanding: Students will have a solid overview of the basic concepts and results of (applied) graph theory, including the main mathematical tools to argue about graphs. Students will have the tools to model and analyze various real-world problems using graphs.
Applying knowledge and understanding: Students will be able to recognize when a problem can be modeled with graphs, and whether the problem can be efficiently solved using standard or slightly adjusted graph-theoretic algorithms.
Making judgements: Students will be able to formulate a given (sub)problem as a graph-theoretic problem, argue why the formulation is correct, and they will be able to judge the feasibility of existing algorithmic solutions.
Communication: Students will be able to explain, in the language of graph theory, how a problem at hand can be modelled and solved.
Learning skills: Students will enhance their study skills such as time management, effective reading, critical thinking and reading, exact and unambiguous writing and formulation of ideas and statements, and reflection on marked (graded) work. Along the way, students will improve general learning skills such as self-motivation, careful listening and giving instructions, and openness to new knowledge.
Reasoning Techniques (Code: KEN2230)

Examiner: Dr. C. Sironi and Prof. Dr. M.H.M. Winands

**Desired Prior Knowledge:** Introduction to Data Science and Artificial Intelligence; Logic.

**Description:** Central in this course is how, based on available data, new knowledge and information can be obtained using reasoning processes. The course will be supported by tutorials, in which the acquired techniques can be put into practice by using Prolog. The following four techniques are discussed:

1. **Reasoning using logic:** syntax, semantics, and inference in first-order logic, situation calculus, forward and backward reasoning, completeness, logic programming with Prolog.
2. **Problem solving using search:** problem types, blind-search methods, informed-search methods, comparison of search methods, games as search problems, minimax, alpha-beta pruning, Monte Carlo Tree Search, chance games, constraint satisfaction problems.
3. **Planning:** planning in situation calculus, representation of states, goals and operators, state space and plan space, algorithms for classic planning.
4. **Reasoning with uncertainty:** uncertainty and probability theory, conditional probability, the Rule of Bayes, semantics of belief networks, exact and approximate inference in belief networks.

**Knowledge and understanding:** Students learn to understand how problems can be represented as logical problems, as search problems, as planning problems or as problems involving uncertainty and get accustomed to reasoning methods to solve problems of all four types mentioned above.

**Applying knowledge and understanding:** Students learn to apply the reasoning methods learned to toy problems and some more complex situations.

**Making judgements:** Students learn to judge which type of knowledge representation is suitable for the problem at hand, and which reasoning technique is suitable to solve the problem at hand.

**Communication:** students can explain the knowledge representation used and reasoning technique chosen to peers and other experts.

**Learning skills:** Students are able to critically reflect on their own and other’s chosen representations and used reasoning methods.

**Study material:**

**Exam:** Closed-book written exam (80% of final grade) and assignments during the course (20% of final grade).

**ECTS:** 4
Machine Learning (Code: KEN2240)

Examiners: Dr. E.N. Smirnov, Dr. E. Hortal Quesada, and Dr. M. Popa

Desired prior knowledge: Introduction to Computer Science 1, Calculus, Linear Algebra, Logic, Probability and Statistics

Prerequisites: None

Description: Machine learning is a major frontier field of artificial intelligence. It deals with developing computer systems that autonomously analyse data and automatically improve their performance with experience. This course presents basic and state-of-the-art techniques of machine learning. Presented techniques for automatic data classification, data clustering, data prediction, and learning include Decision Trees, Bayesian Learning, Linear and Logistic Regression, Recommender Systems, Artificial Neural Networks, Support Vector Machines, Instance-based Learning, Rule Induction, Clustering, and Reinforcement Learning. Lectures and practical assignments emphasize the practical use of the presented techniques and prepare students for developing real-world machine-learning applications.

Knowledge and understanding: After successful completion of the course, students will be able to describe and explain the basic machine learning algorithms. Students will understand the mathematical foundation of machine learning algorithms and how mathematical methods are successfully combined to obtain the variety of machine learning algorithms that are currently available.

Applying knowledge and understanding: Students will acquire the knowledge able to apply, formulate, and validate techniques from machine learning and to apply basic machine learning algorithms on real-life problems. Students will be able to implement machine-learning algorithms in software and to apply existing machine learning software implementation to datasets. Students will have the necessary knowledge to design, implement, and apply data processing systems that autonomously extract information from data, interpret results, and make decisions.

Making judgements: Students learn how to critically analyse real-world problems, to select appropriate machine learning techniques according to the specific problem, and to predict the consequences of their choices. After successful completion of the course, students gain the ability to judge which problems can be solved better and to which extend through the application of machine learning techniques. Students obtain an awareness of and responsibility for ethical and social consequences of developments in and application of machine learning.

Communication: The skills acquired during the course will allow students to present the results of different stages of the application of machine-learning techniques to specialists or non-specialists.

Learning skills: After successful completion of the course, students can analyse, adapt, design, implement, and critically reflect on machine-learning algorithms and tools. Students also obtain the critical fundamental skills and knowledge to study further advanced machine learning techniques in the professional literature.

Study material: Lecture material provided during the lecture.

Recommended literature:
- H. Blockeel, Machine Learning and Inductive Inference (course text), Uitgeverij ACCO, 2012.

Exam: Written “open-book” exam at the end of the course. During the course, students receive several graded assignments that can earn them a maximum bonus grade of 1.0.

ECTS: 4
Linear Programming (Code: KEN2520)

**Examiner:** Dr. S. Kelk.

**Desired Prior Knowledge:** Linear Algebra.

**Prerequisites:** None

**Description:** A linear program is very different to, say, a Java program. It simply consists of a linear objective function (of potentially very many variables) and a set of linear inequalities. The goal is to find values of the variables, which maximize or minimize the objective function, subject to all the inequalities being satisfied. Linear programs - even very large linear programs - can be solved extremely quickly, in both theory and practice. The model is also expressive enough to capture a large number of real-world problems. These two factors explain the fundamental role of linear programming in operations research, computer science, economics, management and many other fields. The course consists of an in-depth study of the simplex algorithm (a standard algorithm for solving linear programs), duality theory, and sensitivity analysis. Examples from practice illustrate the power of the model and teach the student the skill of modelling. Practical aspects of linear programming (e.g. use of software packages for solving linear programs, and integration with languages such as Java) are also considered.

**Knowledge and understanding:** Students will be able to identify which real-world optimization problems can be formulated as linear programs. Students will be able to describe the mathematical foundations of the Simplex algorithm for solving linear programming, and articulate how these foundations impact upon the performance of the Simplex method in practice. Students will recognize the power and importance of duality theory for reasoning about the behaviour of linear programs (in particular with regard to sensitivity analysis). Students will be able to exhibit an awareness of non-Simplex paradigms for solving linear programs (interior-point methods) and be able to recount the importance of the linear programming model in operations research and applied mathematics.

**Applying knowledge and understanding:** Students will be able to 1) translate mathematical models into linear programs, 2) to apply the Simplex method by hand to solve small linear programs, 3) to show how the Simplex method behaves in normal and exceptional cases, 4) to manipulate the algebra underpinning the Simplex method, 5) to combine insights from this algebra and primal-dual relations to make rigorous statements about the (sub)optimality of solutions to linear programs, 6) to argue how small changes to linear programs impact upon their optima (sensitivity analysis), 7) to explain key differences between the Simplex method and interior-point methods, and to 8) leverage linear-to program arguments when developing simple algorithms for combinatorial optimization problems.

**Making judgements:** Students will be able to distinguish between mathematical models that can and cannot be cast as a linear program. Students will be able to contrast and compare the behaviour of the Simplex algorithm with interior-point methods. Students will be able to select, out of a large range of algebraic and duality-based instruments, appropriate tools for making rigorous statements about linear programs.

**Communication:** Students will be able to formulate linear programs and defend their correctness. Students will be able to clearly articulate and defend algebraic and duality-based arguments concerning linear programs.

**Learning Skills:** By the end of the course, students will be able to autonomously and critically reflect upon the appropriateness of the linear programming paradigm for tackling optimization problems arising in practice and be able to assess the correctness of mathematical arguments pertaining to linear programming. Students will be able to identify follow-up literature, which goes beyond the scope of the material presented in the course.


**Recommended literature:** students are beforehand encouraged to refresh their knowledge of: (unique) solutions of systems of linear equations, matrix inversion, and matrix rank.

**Exam:** Written exam and optional weekly bonus exercises (the results of which are added to your exam score, up to 10%).

**ECTS:** 4
Period 2.3

Course title: Project 2-1 (KEN 2300)

Examiners: Dr. C. Sironi, Dr. E. Smirnov, Prof. Dr. M.H.M. Winands, and Dr. M. Bousse
Coordinator: Dr. K. Schneider
Tutors: Dr. M. Musegaas, Dr. K. Schneider

Prerequisites: Students must have passed Project 1-1. Furthermore, the student has to have passed at least two out of the following three courses: Introduction to Computer Science 1, Introduction to Computer Science 2, and Data Structures and Algorithms. This project is a prerequisite for Project 3-1.

Description: Students work on a project assignment in small groups of about six students. The group composition stays the same for the whole project and is announced at the beginning of period 2.1. The students are guided through the project by a fixed tutor. The project assignment is divided into three subtasks (one per period) and is strongly related to the content of the courses from period 2.1 and 2.2. In periods 2.1 and 2.2, the students work on the project, while also having to attend the courses of these periods. They meet their tutor approximately once a week. In period 2.3, the students work three weeks full-time on the project and meet their tutor twice a week.

At the beginning of each period, the students have to hand in a planning for the current phase. At the end of each period, the students have to give a presentation and the source code, presentation, and an overview of who did what need to be uploaded to Canvas. While the presentations at the end of period 2.1 and 2.2 are in front of the examiners and the tutors, the presentations at the end of period 2.3 will additionally be in front of the fellow students. In period 2.3, they furthermore have to hand in a report and attend a product and report examination.

Applying knowledge and understanding: Students will learn to concretize project assignment and construct and maintain a planning. Furthermore, they will learn formulating, selecting and validating models for the game chosen and collect and interpret experimental data with evaluation metrics. Lastly, they will improve their ability to plan and chair meetings, create notes for minutes, work in a team such that the workload is balanced and plan teamwork by setting deadlines and distributing tasks.

Making judgements: After completing this project, students will be able to compare and criticize results, position them in terms of the literature diagnose limitations and formulate a discussion.

Communication: Students will be able to write a scientific paper that: describes the project, explains the methods, summarizes the outcomes, discusses them and makes the conclusions. Students will be able to present and defend project in English and coordinate project progress in project meetings.

Learning skills: Students will be able to reflect on the progress of the project and study relevant literature to solve problems at hand.

Study material: Project manual project 2-1, Maastricht University, DKE.

Assessment: Final grade = 0.15 x grade of phase 1 + 0.15 x grade of phase 2 + 0.7 x grade of phase 3, where grade of phase 3 = 45% grade for product + 35% grade for report + 10% grade presentation + 10% grade project management. Missing project meetings and skill classes will cause an individual grade reduction of the final grade. Furthermore, the examiners may deviate from the group grade for individual students if they either contributed significantly more or less than other group members.

Skill Classes:
LaTeX (period 2.1)
In this skill class, you will learn the basics of LaTeX such as figures, tables, referencing and formulas. At the end of the course, you will be able to write reports and articles with LaTeX.

Systematic search strategies (period 2.1):
In this skill class, you will learn differences between search databases and an effective way of finding necessary references.

Legal aspects of Data Science (period 2.2)
This project skills class will consist out of two sessions. In the first session, you will get an introduc-
tion into the basic principles of the GDPR. By the end of this session, you will be able to grasp the key actors, concepts and obligations of the GDPR, and develop awareness and understanding of the legal requirements you will encounter in your professional career. The general introduction of the aspects of the GDPR is concretized by a number of examples and the session is ended in an interactive Q&A.

In the second session, you will explore the ethical grey zone that exist next to the legal compliance obligations contained in the GDPR. The concepts of digital ethics and accountability are explored along with their limitations. This is done by investigating different real-life cases and scenarios in a lecture.  

ECTS: 6

Period 2.4

Mathematical Modelling (Code: KEN2430)
Examiner: Dr. J. Karel and Prof. Dr. Ir. R. Peeters.
Desired Prior knowledge: Linear Algebra, Calculus, Matlab.
Prerequisite: None.
Description: Mathematical modelling is of great importance for solving practical problems by casting them into a form suitable for the use of mathematical techniques. In this course, a number of basic topics are discussed. First, attention is paid to a framework for mathematical modelling. Then we focus on some widely used model classes from engineering, in particular on the class of linear time-invariant dynamical models. These are described by linear difference equations (in discrete time) or linear differential equations (in continuous time). Alternative model descriptions that are discussed are transfer functions (in the frequency domain) obtained with the z-transform and the Laplace transform respectively; and state-space models, which may or may not involve canonical forms. Some further topics receiving attention are the concepts of stability, sinusoidal fidelity, Bode diagrams, the interconnection of subsystems, and the technique of pole placement by means of state feedback.

The subject matter is clarified through exercises and examples involving practical applications. Also, relevant functionality in Matlab is introduced, which offers a powerful instrument for analysing linear dynamic models.

Knowledge and understanding: Being able to formulate linear dynamical models, state properties and define representations. Identify frequency domain properties of systems and relate them to applications in signal processing.

Applying Knowledge and understanding: Being able to construct elementary mathematical models. Perform model analysis and extract model properties. Employ various model representations and choose the most appropriate one. Compute state-feedback control.

Making judgements: To recognize what are the important aspects to consider when building a mathematical model. Decide on stability of models.

Communication: Being able to convey properties of models to specialists and non-specialists.

Learning skills: Being able to independently find Matlab functionality to solve basic problems in systems theory.

Study material: Lecture notes.
Exam: Written exam and assignments and/or bonus assignments.
ECTS: 4
Human Computer Interaction and Affective Computing (Code: KEN2410)

**Examiner:** Dr. S. Asteriadis

**Desired Prior Knowledge:** Machine Learning, Probabilities and Statistics.

**Course Description:** Human-Computer Interaction (HCI) is the study of interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioural sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface, which includes both software and hardware; for example, characters or objects displayed on a personal computer's monitor and input received from users via hardware peripherals such as keyboard, mouse and web cameras. This course also covers Affective Computing, a new branch of HCI that places emphasis on user emotions and personality. Affective Computing attempts to bring emotions into intelligent interfaces that interact with humans and see how they can have a positive and constructive impact in human-machine interactions.

**Knowledge and understanding:** The course shows guidelines for the design, implementation and evaluation of HCI systems and addresses interaction styles with the user. Students will be able to identify ways to involve user emotion, personality and cognition in the design of an HCI intelligent interface.

**Applying knowledge and understanding:** Students will be able to apply stages of successful user interface design. Formulate user interview techniques, rapid prototyping, design interfaces and conclude on user needs in business-like scenarios.

**Making judgements:** Students will show awareness of how to involve the user in the design procedure and solve HCI problems based on judgement analysis of user evaluations and interviews.

**Communication:** Students will be able to present their prototypes, communicate and defend their results. They will be able to examine existing interfaces and they will ground their findings based on HCI guidelines.

**Learning skills:** Students will be able to describe a course of action for designing human-centric systems, applicable in a variety of social and business practices. They will be able to establish links among Machine Intelligence, Affective Interactions and Statistical analyses in the design of user-centered interfaces. They will be able to use these skills in constructing personalized solutions, quantitatively analyse and apply user needs, design emotion-capturing techniques by using computational models of affect.

**Study material:** Lecture Slides and other sources that will be made available.

**Recommended literature:**
- Coursera video lectures of Scott Klemmer and accompanying slides.

**Exam:** Individual (90% of final grade) and group (10% of final grade) assignments.

**ECTS:** 4

Theoretical Computer Science (Code: KEN2420)

**Examiner:** Dr. G. Stamoulis

**Desired Prior Knowledge:** Introduction to Data Science and Artificial Intelligence, Discrete Mathematics, Data Structures and Algorithms.

**Description:** This course explores the theoretical underpinnings of computing by investigating algorithms and programs casted as language recognition problems. The influence of the theory on modern hardware and software system design is demonstrated. The following subjects will be treated: mathematical foundations, alphabets and languages, finite automata and regular languages, Turing machines, acceptance and decidability, recursive functions and grammars, time complexity classes, NP problems, NP-completeness.
Knowledge and understanding: Students will learn to comprehend the inherent complexity of problems and be able to motivate why some problems are inherently more difficult than others are. They will learn to have insight into how complex problems can be solved efficiently and will be able to classify such problems into a language hierarchy and complexity classes. Furthermore, students will be able to apply the tools needed for such classification.

Applying knowledge and understanding: Students will be able to apply the theory learned to solve small-scale problems.

Making judgements: Students will learn to judge which problems are decidable and efficiently solvable and to judge which technique is suitable to solve the problem at hand.

Communication: The knowledge representation used and technique from complexity theory chosen must be easily understandable by peers and others experts.

Learning skills: The student will learn to reflect on own one’s and other’s thoughts on complexity and solvability of problems.


Exam: Written exam; during the course the students will receive three assignments, that, if they receive a sufficient grade, may earn them up to a total of one bonus point.

ECTS: 4

Period 2.5

Philosophy & Artificial Intelligence (Code: KEN2120)

Examiner(s): Dr. D. M. Cressman.

Prerequisites: None.

Description: One of the characteristics of scientific knowledge is the translation of natural phenomena into quantitative or mathematical data – the book of nature, Galileo wrote, is written in the language of mathematics. Over the course of the twentieth and twenty-first century, this desire to understand the world through the logic of mathematics has been extended beyond the natural world to include such things as human consciousness, learning, and intelligence. Indeed, the foundation of what is called ‘artificial intelligence’ is the pursuit of replicating human consciousness and intelligence through mathematical models and formulas. In this course we will examine these issues from a philosophical perspective, beginning with a basic overview of the philosophy of science with an emphasis on quantification and then moving on to study philosophical issues that have developed out of the pursuit of artificial intelligence. We will begin with classic thinkers in the field like Alan Turning, Hubert Dreyfus, and Joseph Weizenbaum and continue through to contemporary philosophical studies of cutting edge attempts to develop types of machine learning that aim to mimic human forms of learning.

Knowledge and understanding: At the conclusion of this course, students should be able to demonstrate knowledge of the following topics through written essays:

- The history of computing and artificial intelligence
- The history and philosophy of scientific knowledge with an emphasis on Kuhn’s theories of scientific paradigms
- Historical and philosophical theories of technology and society
- The philosophical presuppositions of artificial intelligence

Applying knowledge and understanding: Students will be able to draw upon both lectures and readings to write an essay that exhibits critical reflections on conventional and naïve notions of instrumentalism, technological determinism, and functionalism by persuasively arguing for a contextual approach that highlights the contingency and flexibility of design and meaning.
Making judgements: Students are asked to select relevant passages from texts that contribute to the argument that they make in the essay. This will be graded. In tutorials, students are asked to make decisions about specific problems (i.e. self-driving cars, Turing tests). This is not graded.

Communication: During tutorials, students present their work orally to their classmates.

Learning Skills: Students will be able to articulate and solve problems in groups. Students will also be expected to engage with a number of theories concerning computation and artificial intelligence through different texts and will be asked to reflect upon and critique these theories.

Study material: Selected texts will be made available.

Exam: Take home exam (essay).

ECTS: 4

Simulation and Statistical Analysis (Code: KEN2530)

Examiner: Dr. J. Karel
Prior Knowledge: Knowledge: Probability & Statistics, Calculus, Matlab, and Java.
Prerequisite: None.
Description: Mathematical simulation is concerned with studying processes and systems. Uncertainty can be an important factor and has to be modelled properly. After modelling a complex system, various scenarios can be simulated, using Monte Carlo simulation, to gain insight. The results need to be properly interpreted and uncertainty has to be reduced. The modelling, implementation, analysis and technical aspects will be discussed as an introduction in this field. Emphasis will be on discrete event simulation and the statistical analysis of the output of simulation studies, where topics are: modelling, Poisson processes, random number generators, selecting and testing input distributions, generating random variates, statistical analysis of experiments, comparing experimental results and variance reduction. Practical exercises will be used to place the techniques in context.

Knowledge and understanding: Define concepts of simulation, discrete event simulation and statistical inference. Explain techniques underlying mathematical simulation. Recall and explain methods for analyzing experimental results and efficient simulation including their assumptions, justify why they are important, and match them to simulation design.

Applying Knowledge and understanding: Being able to model a system in a structured manner, to design and implement simulators for systems, and to collect data from these simulations. In addition, you will be able to employ techniques underlying mathematical simulation and apply methods from statistics for analyzing experimental results.

Making judgements: Being able to choose and motivate alternative techniques underlying mathematical simulation. Choose, motivate and contrast methods for analyzing experimental results and efficient simulation.

Communication: Being able to convey the phases of a specific simulation study to non-experts. Being able to explain the assumptions and choices made when analyzing experimental data to experts and non-experts.

Learning Skills: The ability to independently learn to handle large-scale simulation. To identify shortcomings in data analysis.

Study material: Simulation Modeling and Analysis (5th edition) - Averill Law

Exam: Written exam and assignments and/or bonus assignments.

ECTS: 4
Natural Language Processing (Elective) (Code: KEN2570)

Examiner: Dr. J. Niehues

Tutors(s): None.

Desired Prior Knowledge: Introduction to Computer Science 1 and 2, Probability and Statistics, Machine Learning

Prerequisites: None

Description: Watson won Jeopardy. Siri can tell me when I need an umbrella. But how do they work? Over the past decade, Natural Language Processing (NLP) was revolutionized by statistical, probabilistic and machine learning methods. NLP addresses fundamental questions at the intersection of human language and machine learning. How can computers acquire, understand and produce language? How can computational methods give us insight into observed human language phenomena? How can we make sense of the vast amounts of information available online in free, unstructured form? In this course students will learn how computers can learn useful text/language representations and how different tasks (language modelling, text classification, information extraction, sequence labeling, etc.) can be used for solving different complex problems (spelling correction, spam detection, search engine design, opinion analysis, summarization, question-answering, etc.). Open NLP problems (such as evaluation or interactive dialogue systems) and the effect of deep learning on NLP will be discussed.

Knowledge and understanding: By the end of the course, students are able to acquire the basic text and language processing aspects. Furthermore, students are able to describe basic NLP problems, tasks and methods.

Applying knowledge and understanding: Students are able to demonstrate how to tackle a text/language problem and to formulate, design and implement a NLP system. Students are able to suggest when a problem’s complexity requires an NLP solution.

Making Judgements: Students are able to pose questions and define problems in different domains (e.g. social sciences) and contexts (e.g. business) that include language/text data. Furthermore, students are able to judge which tools are applicable for solving these problems and to decide a course of action in accordance with ethical and social consequences.

Communication: Students are able to outline an approach in real organizational problems, which require NLP and are able to demonstrate, present and communicate a solution to a NLP problem

Learning Skills: Students are able to master and choose the appropriate basic programming tools for NLP and are able to follow up on literature that will allow them to build complete NLP models.

Study material: Handouts

Recommended Literature:
2) Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA. 1999

Exam: Practical individual assignments (30%) + Group Project (20%) + Open-Book Written Exam (50%).

ECTS: 4
Examiner: Dr. A. Briassouli  

Desired prior knowledge: Calculus, Linear Algebra, Machine Learning.  

Prerequisites: None.  

Course Description: Image and video processing is everywhere around us, in smartphones, robotics, medicine, security systems, microscopy, remote sensing, video games, travel, shopping, environmental management and many other applications. Image and video processing is based on principles of signal processing, extended to multiple dimensions. In this class students will have a short introduction to basic 2D signals and systems, sampling, convolution. Color domain processing in different spaces and its relevance to our visual perception system will be presented. We will learn about linear and non-linear filtering in the spatial and frequency domains (Fourier, DCT), their relation and applications like enhancement, noise estimation and removal, compression, restoration. Compression standards for image and video and their relevance to frequency transformations will be presented. Video analysis will include methods for motion estimation, segmentation and introduction to action recognition, and video standards. Lab examples and homeworks will accompany the classes.  

Knowledge and understanding: Students will gain an in-depth knowledge of image and video processing methods used all around us by understanding the underlying theoretical foundations and obtaining insights into their role in numerous applications. They will understand the mathematics and algorithms underlying image and video analysis methods and see their results in practice through mini projects.  

Applying knowledge and understanding: The students will be able to immediately apply basic image and video processing concepts to real world problems in the labs. They will also have the opportunity to further demonstrate their understanding by implementing them in mini projects.  

Making judgements: By understanding fundamental signal/image processing principles and seeing them work in practice, the students will be able to understand how and where to implement and develop image and video analysis algorithms, as well as build complex systems using and extending them.  

Communication: Assessment will be based on mini projects with topics on image/video processing chosen by the instructor (the students themselves are also free to suggest subjects relevant to the class). The goal will be for them to learn to carry out independent work, solve realistic problems based on the class material, as well as effectively communicate their motivation and results both to a general and expert audience by providing interesting demos, presentations and clearly structured reports.  

Learning Skills: Students will be able to carry out basic image and video analysis algorithms, understanding and implementing the mathematics behind them. They will solve small research questions within mini projects and will become familiarized with the newer toolboxes and libraries, mostly in – but not restricted to - Matlab and Python.  

Study material: Lecture slides and provided material. Gonzalez & Richard E. Woods, “Digital Image Processing”.  


Exam: Three mini projects (100%).  

ECTS: 4
Period 2.6

Course title: Project 2-2 (KEN 2600)

Examiners: Dr. A. Briassouli and Dr. J. Niehues

Coordinator: Dr. K. Schüller

Tutors: Dr. K. Schüller and Dr. M. Musegaas

Prerequisites: Students must have passed Project 1-2. Furthermore, the student has to have passed at least two out of the following three courses: Introduction to Computer Science 1, Introduction to Computer Science 2, and Data Structures and Algorithms. This project is not a prerequisite for another project / course.

Description: Students work on a project assignment in small groups of about six students. The group composition stays the same for the whole project and is announced at the beginning of period 2.4. The students are guided through the project by a fixed tutor. The project assignment is divided into three subtasks (one per period) and is strongly related to the content of the courses from period 2.4 and 2.5. In periods 2.4 and 2.5, the students work on the project, while also having to attend the courses of these periods. They meet their tutor approximately once a week. In period 2.6, the students work three weeks full-time on the project and meet their tutor twice a week.

At the beginning of each period, the students have to hand in a planning for the current phase. At the end of each period, the students have to give a presentation and the source code, presentation and an overview of who did what need to be uploaded to Canvas. While the presentations at the end of period 2.4 and 2.5 are in front of the examiners and the tutors, the presentations at the end of period 2.6 will additionally be in front of the fellow students. In period 2.6, they furthermore have to hand in a report and attend a product and report examination.

Applying knowledge and understanding: Students will learn to concretize project assignment and construct and maintain a planning. Additionally, they will learn formulating, selecting and validating models for a concrete problem at hand and to collect and interpret data with evaluation metrics. Lastly they will improve their ability to plan and chair meetings, create notes for minutes, work in a team such that the workload is balanced and plan teamwork by setting deadlines and distributing tasks.

Making judgement: After completing this course successfully, students will be able to compare and criticize results, position them in terms of the literature; diagnose limitations and formulate a discussion.

Communication: Students will be able to write a scientific paper that: describes the project, explains the methods, summarizes the outcomes, discusses them and makes the conclusions. Furthermore, student will be able to present and defend project in English. Coordinate project progress in project meetings.

Learning skills: Students will learn to reflect on the progress of the project and study relevant literature to solve problem at hand.

Study material: Project manual project 2-2, Maastricht University, DKE.

Assessment: Final grade = 0.15 x grade of phase 1 + 0.15 x grade of phase 2 + 0.7 x grade of phase 3, where grade of phase 3 = 30% grade for product + 50% grade for report + 10% grade presentation + 10% grade project management. Missing project meetings and skill classes will cause an individual grade reduction of the final grade. Furthermore, the examiners may deviate from the group grade for individual students if they either contributed significantly more or less than other group members.

Skill Classes:

Academic Writing (period 2.4)
You will receive feedback on your project report draft.

Academic Presentation (period 2.5)
In this skill class, you will practice your presentation and you will receive feedback on the presentation style, your slide layout, your pronunciation, and your language.

ECTS: 6
2.3 Curriculum of the Third Year of the Bachelor’s Programme

The first semester of the third year allows you to make your own selection of subjects in the field of artificial intelligence, data science, applied mathematics and computer science, the core areas of the study of Data Science & Artificial Intelligence. In period 1 and 2, you can choose 6 out of 12 optional courses. Alternatively, students can choose to study the first semester of the third year at a university abroad. Please check the Study abroad guide for more info. Other options include following courses at other faculties. The first semester of year 3 has the same structure in the first and the second year; there are two periods of eight weeks and one period of four weeks. There is also a project in period 3.3.

<table>
<thead>
<tr>
<th>Year 3</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 3.1</td>
<td></td>
</tr>
<tr>
<td>Parallel Programming (KEN3235)</td>
<td>4</td>
</tr>
<tr>
<td>Semantic Web (KEN3140)</td>
<td>4</td>
</tr>
<tr>
<td>Game Theory (KEN3130)</td>
<td>4</td>
</tr>
<tr>
<td>Robotics and Embedded Systems (KEN3236)</td>
<td>4</td>
</tr>
<tr>
<td>Computer Security (KEN2560)</td>
<td>4</td>
</tr>
<tr>
<td>Digital Society</td>
<td>4</td>
</tr>
<tr>
<td>Project 3-1 (**)</td>
<td></td>
</tr>
<tr>
<td>Period 3.2</td>
<td></td>
</tr>
<tr>
<td>Large Scale IT and Cloud Computing (KEN3239)</td>
<td>4</td>
</tr>
<tr>
<td>Logic for Artificial Intelligence (KEN3231)</td>
<td>4</td>
</tr>
<tr>
<td>Recommender Systems (KEN3160)</td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Bio-Informatics (KEN3440)</td>
<td>4</td>
</tr>
<tr>
<td>Software and Systems Verification (KEN3150)</td>
<td>4</td>
</tr>
<tr>
<td>Introduction to Quantum Computing (KEN3241)</td>
<td>4</td>
</tr>
<tr>
<td>Project 3-1 (**)</td>
<td></td>
</tr>
<tr>
<td>Period 3.3</td>
<td></td>
</tr>
<tr>
<td>Project 3-1 (KEN3300)</td>
<td>6</td>
</tr>
<tr>
<td>Period 3.4</td>
<td></td>
</tr>
<tr>
<td>Data Analysis (KEN3450)</td>
<td>4</td>
</tr>
<tr>
<td>Operations Research Case Studies (KEN3410)</td>
<td>4</td>
</tr>
<tr>
<td>Intelligent Systems (KEN3430)</td>
<td>4</td>
</tr>
<tr>
<td>Period 3.5-3.6</td>
<td></td>
</tr>
<tr>
<td>Bachelor’s thesis (KEN3500)</td>
<td>18</td>
</tr>
</tbody>
</table>

* Third year students choose three electives per period out of the optional courses during period 1 and 2
** Project 3-1 will start in period 3.1 and 3.2 with weekly meetings. The credits for the project will become available at the end of period 3.3. For each period, we will give a short explanation of the various courses. Before the start of each period, the students will receive detailed information about the content, the study material, the teaching form, the schedule, and the examination method.
Parallel Programming (optional course) (Code: KEN3235)

Examiner: Prof. Dr. H. Pflug, B. Küppers M.Sc.

Prerequisites: Introduction to Computer Science 1 and 2, Data Structures and Algorithms.

Description: Parallel programming introduces the students to the paradigm of parallel computing on a computer. Nowadays almost all computer systems include so-called multi-core chips. Hence, in order to exploit the full performance of such systems one needs to employ parallel programming.

This course covers shared-memory parallelization with OpenMP and java-Threads as well as parallelization with message passing on distributed-memory architectures with MPI. The course starts with a recap of the programming language C followed by a brief theoretical introduction to parallel computing. Next, the course treats theoretical aspects like MPI communication, race conditions, deadlocks, efficiency as well as the problem of serialization. This course is accompanied by practical labs in which the students have the opportunity to apply the newly acquired concepts. After completing this course students will be able to write parallel programs with MPI and OpenMP on a basic level, and deal with any difficulties they may encounter.

Knowledge and understanding: Students recall the basic concepts for parallel programming and recognize important parallelization patterns.

Applying knowledge and understanding: Students are able to write parallel software code using MPI, OpenMP, and Java Threads.

Communication: Students are able to explain why a specific pattern is adequate for a given problem.

Learning Skills: Students are able to study autonomously the literature describing parallel programming in order to comprehend important details and problems of the field.

Study material: Course notes and several codes will be provided online.

Recommended literature: Parallel programming with MPI; Peter Pacheco; Morgan Kaufmann (1996); (a very early revision is available online)

Exam: Written exam.

ECTS: 4
Semantic Web (optional course) (Code: KEN3140)

Examiner: Prof. Dr. M. Dumontier

Desired Prior Knowledge: Logic

Description: Most of the information available on the World Wide Web (WWW) is not directly understandable for computers. For instance, web pages are designed for human readability. Computer programs have difficulty in interpreting the information presented on web pages. The focus on human readable information introduces restrictions on what computer programs can do to support human users in tasks such as:

- finding information
- buying goods
- making travel plans

The Semantic Web should eliminate these restrictions by separating the content of what is presented on a web page from the way it is presented. In recent years, the focus has shifted to providing data, independent of webpages (for example: Linked Open Data (LOD)). Ontologies are used to provide a shared conceptualization of information. Ontologies form the basis of the Semantic Web, Knowledge Based System, Databases, etc., and they play an important role in data exchange and interoperability in many domains. Ontologies are applied in the bio-medical domains, in data mining applications, in Linked Open Data (LOD), in websites based on semantic technology, etc.

Since ontologies are intended to be shared between different systems, defining an ontology is a challenging task. This course will focus on the standards the World Wide Web Consortium (W3C) is defining in order to realize the Semantic Web. The course also addresses the underlying knowledge representation formalisms of the current semantic web standards. Moreover, the course will address the engineering principle of crating an ontology. Note that the course does not address standards for making websites.

Knowledge and understanding: Making the student familiar with the developments and standards of the Semantic Web. The student will get insights in semantic web standard, such as RDF, RDFa, SPARQL and OWL2. Moreover, the students will get some basic insight in the semantics of RDF and the Description Logic underlying OWL. Finally, the student will be made familiar with the ontology development process, and criteria for evaluating an ontology. The student should understand the role of upper ontologies and ontology design patterns, as well as the philosophical choices they represent.

Applying knowledge and understanding: The student should be able to build applications using semantic web standards such as RDF, RDFa, SPARQL and OWL2. The student should also be able to develop an ontology for an application domain.

Making judgements: The student should be able to judge whether and how semantic web standards can be applied in applications. The student should also be able to judge the quality of an ontology.

Communication: The student should have sufficient understanding of the Semantic Web and its standards in order to explain why and how an application should be set up using semantic web standards. The student should also be able to explain and defend the choices made in the ontology engineering process.

Learning skills: The student should be able to study the literature about semantic web developments.

Study material:
- Syllabi and scientific papers about ontology engineering.


Examination: Practical exercises and a written exam at the end of the course. The grade is for 70% determined by the written exam and for 30% by practical assignments. Participation in the practical is required for receiving a grade.

ECTS: 4
Game Theory (optional course) (Code: KEN3130)

**Examiner:** Prof. Dr. F. Thuijsman

**Prerequisites:** Discrete Mathematics, Linear Algebra

**Description:** We introduce the field of Game Theory. Game Theory is the mathematical study of problems, called games, that involve two or more decision makers, called players, who each have their own individual preferences over the possible outcomes. In a game, each player always aims to maximize his individual payoff and chooses his actions accordingly. These actions may be probabilistic or deterministic, depending on the situation. Meanwhile he reasons logically about actions that might be taken by the other players. A basic difference exists between strategic and non-strategic models. Both types of models and their solution concepts will be discussed. Issues like value, fairness, manipulations, threats, optimality and rationality will be addressed.

**Knowledge and understanding:** Students can recognize and classify the main types of games, i.e. cooperative games, strategic games, bipartite matching problems, and formulate the main solution concepts value, optimal strategies, Nash- and correlated equilibrium, as well as a number of algorithms to calculate these.

**Applying the use of knowledge and understanding:** Students can calculate solutions for the different types of games

**Making Judgements:** Students can explain advantages and disadvantages of different solution concepts. They are able to judge correctness of solutions presented

**Communication:** Students can explain and defend correctness of their solutions

**Learning Skills:** By the end of the course, students will be able to autonomously and critically reflect upon the pros and cons of different types of games for modelling competition and cooperation. This includes considerations on the computational aspects with respect to different solution concepts.

**Study material:** Lecture notes.

**Examination:** There will be a closed book written exam at the end of the course.

**ECTS:** 4
Robotics and Embedded Systems (optional course) (Code: KEN3236)

Examiner: Dr. R. Möckel.

Desired prior knowledge: Calculus, Linear Algebra, Machine Learning.

Prerequisites: Introduction to Computer Science 1 and 2.

Description: Nowadays, a variety of products require that algorithms from data science and artificial intelligence are adapted to and implemented in robotic and embedded systems. Applications that heavily rely on intelligent robotic and embedded systems include self-driving cars, autonomous drones, intelligent industrial robots in (semi-)autonomous factories, smart phones, intelligent medical devices, and distributed intelligent embedded devices in smart homes.

In this course, students receive an introduction to the fields of robotics, embedded systems, and real-time control. Students obtain an overview of state-of-the-art intelligent robotic and embedded systems in academia and industries. Students gain hands-on experience in programming embedded robotic systems using embedded processors and a modular robotic system developed at DKE. Students learn about communication standards for embedded systems, sensors, and actuators. Student practise and strengthen their expertise in data science and knowledge engineering by applying mathematical methods for controlling robotic systems: They study control techniques including PID control, forward and inverse kinematics as well as locomotion control and learning using central pattern generators. The course concludes with a robot competition where students build and program robots using a modular robotic system.

Knowledge and understanding: Students obtain knowledge in designing, building, and programming robotic and embedded systems. Students learn how to apply mathematical concepts like dynamic systems for controlling robotic systems in real-time. Students further obtain knowledge about sensors and motor control and study the application of machine learning and mathematical methods for learning and optimizing control parameters. Students receive training in the programming language C - the most popular languages for programming microcontrollers.

Applying knowledge and understanding: After successful completion of this course, students can analyse, apply, implement, and validate control techniques in embedded and robotic systems with and without real-time constraints. Students can apply techniques from machine learning, search, and optimisation to obtain parameters for embedded control systems as required in many professional academic and industrial applications.

Making judgements: Students learn to judge where real-time systems are required and embedded systems can be beneficial. Students further learn to critically analyse the use of robotic systems in a variety of scenarios and to make design choices for robotic and embedded systems. By introducing students to a variety of state-of-the-art robotic systems, the course lays the foundation so that students can process professional literature in robotics and embedded systems.

Communication: Students will be able to 1) discuss robotic and embedded systems professionally and critically, 2) plan, discuss, implement, and validate projects in robotics and embedded, 3) present the results of project assignments in form of video, and to 4) critically analyse and explain control techniques for robotic systems to a general and professional audience.

Learning Skills: Students are able to autonomously and critically reflect upon the abilities and limitations of robotic and embedded systems in order to keep up with new developments in the field. Students can further assess the capabilities and limitations of their own solutions to a control or machine learning problem in robotics, and to identify follow-up literature, which goes beyond the scope of the material presented in the course.

Study material: Course material will be provided during the lectures.

Exam: The final course grade is 80% of the final written “closed-book” exam grade plus 20% of the assignments grade.

ECTS: 4
Computer Security (optional course) (Code: KEN2560)

**Examiner:** B. Küppers, M.Sc.

**Tutor:** B. Küppers, M.Sc.

**Desired Prior Knowledge:** Introduction to Computer Science (1 & 2), Data Structures and Algorithms, Software Engineering, Databases

**Description:** Computer security is the process of securing information systems against unauthorized access. As information systems have become mandatory in the modern world, coupled with the increased frequency of security incidents, organizations now recognize the need for a comprehensive security strategy. The course will introduce a wide range of topics in computer security and online privacy. The main objective of the course is to cultivate a security mind set by discussing various attack techniques and defenses. The topics we will explore are information security (cryptography, cryptoanalysis), software security and network security, as well as designing secure systems. The class will consist of lectures in which several computer security issues will be discussed. In parallel, there will be bonus assignments where the students will have to solve some of the most important issues we discussed in classroom.

**Knowledge and understanding:** Students will be able to explain the fundamental concepts of computer security, such as the principles of a secure system and the potential attacks that can compromise it. They will also learn to recognize the various aspects that play a key in applying those principles to real world scenarios, and avoid common mistakes that can introduce vulnerabilities to their systems.

**Applying Knowledge and understanding:** Students will be able to apply computer security fundamentals to real world scenarios.

**Making judgements:** By understanding the fundamentals of computer security and by realizing their assignments, the students will be able to understand and avoid mistakes when designing a system. In principle, after completing this course, they will be able to design and develop secure systems on their own.

**Communication:** Students will be able to explain the principles of computer security to specialists and non-specialists. They will be able to explain why the design of a system is secure or not and how the system can be improved.

**Learning skills:** Students will be able to read and interpret scientific literature on computer security that goes beyond the scope of the course, and independently design and implement secure systems for real-world applications.

**Recommended material:** material provided electronically on Student Portal/digital learning environment.

**Exam:** Assignments and Project.

**ECTS:** 4
Digital Society (optional course) (Code: KEN3111)

Examiner: Dr. T. Frissen
Prerequisites: None
Desired prior knowledge: None

Description: Digitalization has a profound impact on our society. We can observe changes in different areas. What digital technologies do, what they look like and how they relate to each other is not identical worldwide, but dependent on local practices as well. Usually new technologies are understood as innovation and progress: and indeed, digital technologies improve a broad range of domains, such as healthcare or education. New possibilities as e.g. participation in our digital cultures arise but also new inequalities, as the access and competences needed for participation are not evenly distributed and the platforms that allow for participation also harbour new mechanisms of control and surveillance. The pace and diversity of these developments ask for continuous investigation and reflection. It requires work to shape and use technologies in ways that contribute to the public good. Moreover, digital technologies have also led to highly problematic developments such as electoral manipulation, fake news and algorithmic discrimination.

Technological developments are often conceived as predefined or given. Does a society’s technology drive the development of its social structure and cultural values? Scholars in science and technology studies have shown that technology and society are deeply intertwined. Technology is inherently social. Technologies are shaped by people; they emerge and are embedded in social practices. The aim of this course is to investigate the consequences of digitalization for our society/societies. These consequences have been differently valued: participation vs. exploitation of users, innovation as enhancement vs. challenge, ethics and techno-moral change vs./and sustainability. We will discuss digitalization from

• a social perspective when we read about digital participation and how technology and society are intertwined
• a political perspective when we discuss activism, digital citizenship but also problems of manipulation and verification (as in the case of fake news and deep fakes)
• a cultural perspective when we analyze imaginaries and discourses around innovation of technology and promises being made
• a legal perspective when we discuss privacy and the attempts to adapt privacy laws
• an ethical perspective when we discuss design decisions, privacy but also techno-moral change and questions of environment and sustainability.

The course is structured in the following way:
Transformations
(digital participation, digital citizenship, data-activism)
Imaginaries
(innovation and techno-moral change)
Disruptions
(fake news and deep fakes, sustainability and e-trash)

Knowledge and understanding: Students acquire knowledge on the impact of digitalization on society.
Applying knowledge and understanding: Students learn to understand the interrelation between digital technology and sociality.
Making judgements: Upon completion of the course, students can reflect on ethical challenges related to digitalization.
Communication: Students are able to communicate central topics related to digitalization to an audience of non-IT-experts (e.g. the debate will bring students from the FASoS BA Digital Society and DKE students following this course together to train both groups to communicate topics related to digitalization from a social science and IT perspective).
Learning Skills: Students are able to reflect critically in written form on a topic related to the digital society but also to do so orally in a presentation and debate.
Study material: The literature (provided via the reference list of the library).
Exam: (group) presentation in class (1-3 students) per task (25% of the final grade), 2 short academic papers of 1500 words each (2x25% of the final grade) and participation in a final debate (25% of the final grade).
If a resit is needed for the (group) presentation, the presentation will be given via video (e.g. Zoom or Skype). If a student needs to resit the papers they can be rewritten and improved based on the comments of the tutor. If a resit is needed for the debate (in case a student does not show or participate in the debate), the student can write a 1500-word paper on the content of the debate instead.
ECTS: 4

Period 3.2

Large Scale IT and Cloud Computing (KEN3239)
Examiner: Dr. T. Eifert, B. Küppers, M.Sc.
Desired Prior Knowledge: Introduction to Computer Science 1, Databases
Prerequisites: none
Description: The course offers a comprehensive introduction to the field of scalable IT systems, so-called “Big IT”, and cloud computing. After a technical introduction to the available methodologies of setting up and running scalable systems, use cases are presented. These use cases emphasize the correlation of the processes and requirements of large institutions and possible technical solutions. A special focus is put upon the question which technological platform is best used for which use case as well as process aspects of scaling. Security aspects specific to cloud computing are discussed along the use cases. Cloud computing, as a special case of scalable IT, is discussed in detail. Different cloud providers are presented and evaluated in the context of university requirements, i.e. requirements posed by research and teaching processes.
Knowledge and understanding: students acquire an overview of existing technologies for scalable systems, and specific security requirements for the different use cases.
Applying knowledge and understanding: Students are able to understand scalability and are able to set up and use a scalable IT system. In addition, students are able to evaluate high scalable IT solutions in terms of benefits and security risks.
Making judgements: Students are able to analyze the requirements of a specific use case and can decide which technology is best used for that case of application.
Communication: students are able to communicate about scalable IT systems and specific security requirements.
Learning skills: Additionally, students are able to analyse the interdependencies between large organizations, processes and IT solutions - taking into account security-related aspects - and to design suitable solutions using cloud offerings.
Study material: Lecture notes
Exam: Assignments and Project
ECTS: 4
Logic for Artificial Intelligence (optional course) (Code: KEN3231)

Examiner: Dr. Ir. Ing. N. Roos.

Desired prior knowledge: Knowledge of propositional and predicate logic.

Prerequisites: The first year bachelor course: Logic.

Description: Logics form the formal foundation of knowledge representation and reasoning, which is a fundamental topic in Artificial Intelligence. Logics play a role as an analysis aid and as a knowledge-representation formalism. Moreover, the semantics of logics enables us to evaluate the intended meanings of knowledge representation formalisms, and the correctness and completeness of reasoning processes.

Humans make assumptions in their day-to-day reasoning. Examples of reasoning with assumptions are: common sense reasoning, model-based diagnosis, legal argumentation, agent communication and negotiation, and so on and so forth. The assumptions humans use in their reasoning may be incorrect in the light of new information. This implies that conclusions may have to be withdrawn in the light of new information. Therefore this form of reasoning is called non-monotonic reasoning and the underlying logics are called non-monotonic logics.

The course will cover model-based diagnosis as an application of reasoning with assumption, standard logics extended with defeasible rules, argumentation systems, the semantics of reasoning with assumptions and defeasible rules, and closure properties of the reasoning systems.

Knowledge and understanding:
• The student should be able to describe non-monotonic logics and argumentation systems.
• The student should be able to identify the logic underlying specific forms of knowledge representation.
• The student should be able to describe and discuss the semantic of non-monotonic logics.

Applying knowledge and understanding:
• The student should be able analyze important properties of practical formalism's for knowledge representation and reasoning.
• The student be able to apply non-monotonic logics and argumentation systems to practical problems.

Making judgements:
• The student should be able to judge whether specific knowledge representation formalism's are able to represent the intended meaning of the knowledge to be represented.
• The student should be able to analyze whether conclusions derived from a knowledge representation are correct and complete.

Communication:
• The student should be able to explain how logic can be used as a tool for analyzing a knowledge representation problem.
• The student should be able to explain issues involved in the handling assumptions in a knowledge-representation.

Learning skills:
• The student should be able to study autonomously the literature describing the applications of logics for knowledge representation and reasoning.

Study material: Syllabi.

Recommended Literature: A syllabus and scientific literature.

Examination: Written exam at the end of the course. A bonus of 1.0 point can be earned by a series of bonus assignments.

ECTS: 4
Recommender systems (Code: KEN3160)

Examiners: Prof. Dr. N. Tintarev.

Required Knowledge: Machine Learning

Desired Prior Knowledge: Natural Language Processing, Human Computer Interaction & Affective Computing

Recommender systems play an important role in helping to mediate many of our everyday decisions and choices, including the music we listen to, the news that we read, and even the people that we date. They do this by learning from our past interactions, inferring our interests and documenting our preferences. To make the right suggestions at the right time recommender systems must not only understand our preferences but also our current needs and perhaps our immediate intent. Thus, the core focus of most recommender systems is devoted to profiling users and matching items based on these profiles and current context.

Much of the research to date on recommender systems has focussed on the engineering and evaluation of core recommendation algorithms. Researchers have developed a variety of approaches to harness different forms of preference data in the pursuit of more accurate recommendations. For example, researchers have used simple ratings for collaborative, rich meta-data for content-based methods, and even the opinions and sentiment expressed within user-generated reviews. When evaluating recommender systems, there has been a heavy emphasis on measuring the accuracy of suggestions, or the error of predictions. However, in practice it is important to consider evaluation metrics beyond accuracy, such as diversity, novelty, and serendipity. This in turn has led to increased attention being given to the nature of the interactions between users and recommender systems, and the influence that the user interface and interaction style can have on user behaviour and the overall recommendation experience. This course focuses on:

- Non-personalized and Stereotype-based Recommender Systems
- Classical recommender systems algorithms, e.g., Content-based Filtering, Collaborative-based Filtering
- Offline Evaluation e.g., protocols, criteria, metrics
- User-centered evaluation
- Interfaces and interaction in Recommender systems, e.g., explanations and conversational recommender systems
- Ethics, bias, and fairness in recommender systems
- Advanced methods, e.g., Matrix Factorization, Hybrid recommenders, Learning-to-Rank

Knowledge and understanding: Students will be able to explain concepts from recommender systems, such as the difference between different recommendation methods and can identify advantages and limitations of these methods. Students will also be able to explain one advanced method or topic suitable for progression to a Master level program in Data Science or Artificial Intelligence.

Applying knowledge and understanding: Students will be able to apply ideas, methods, and tools for recommender systems that are suitable for a given domain. Students will be able to solve problems and design analytically, to comprehend (design) problems and abstract their essentials, to construct and develop logical arguments with clear identification of assumptions and conclusions. Students will develop the ability to transpose academic knowledge and expertise into (inter)national societal, professional and business contexts.

Making judgements: Students will gain acquaintance with the standards of academic criticism. Students will develop an awareness of, and responsibility for ethical, normative and social consequences of developments in science and technology, particularly resulting from Data Science and Artificial Intelligence.

Communication: Students will develop academically and internationally appropriate communicative skills, i.e., the ability to give effective oral presentations, both formally and informally, and understand and offer constructive criticism of the presentations of others.

Learning skills: Students will be able to reflect on their own working methods, and own readiness to take the necessary corrective action.

Study material: Course notes, required reading of scientific articles.


Exam: “Reviewing” Scientific Papers (15%) + practical group assignment (30%) + exam (55%)

ECTS: 4
Introduction to Bio-Informatics (Code: KEN3440)

Examiner: Dr. R. Cavill.

Desired Prior Knowledge: Introduction to Computer Science 1, MatLab.

Prerequisites: None.

Description: This course presents a general introduction to the fundamental methods and techniques of bioinformatics in biomedical and biological research. The objective is that the students will acquire a general understanding of bioinformatics methods at the algorithmic level and will therefore be able to read and understand publications in this field, and – to some extent – apply their knowledge to concrete biological problems. This relates to the major areas of bioinformatics like sequence alignment, phylogenetic analysis, gene finding, and omics data analysis. This course consists of a series of closely related lectures and computer classes, based on relevant case-studies using real data. In the lectures the main theoretical aspects are presented. In the computer practicals, the students work to analyse real data using the techniques they have encountered. By extensively exploring the case study, the students acquire a thorough understanding about the subject.

Knowledge and understanding: Students should be able to perform common analyses on both sequence data and numeric data from omics experiments. This includes sequence alignment, building phylogenetic trees, applying hidden Markov models, detecting differentially expression and performing pathway analysis.

Applying knowledge and understanding: For all the above topics students should be able to demonstrate the algorithms on paper with simple examples and apply the algorithms appropriately on realistic datasets using a computer.

Making judgements: After successful completion of the course, students will be able to judge the use, quality, and correctness of different bioinformatics algorithms and results.

Communication: After this course students will be able to explain the algorithmic bases of bioinformatics problems and interact with biologists to provide recommendations of analysis approaches in the situations studied.

Learning Skills: After successful completion of the course students will be able to independently read bioinformatics literature to further their knowledge.


Exam: Written exam (50%) + assignments (50%).

ECTS: 4
Software and Systems Verification (optional course) (Code: KEN3150)

**Examiner:** Dr. Pieter Collins  
**Tutor:** Dr. Pieter Collins

**Desired prior knowledge:** Reasoning Techniques, Theoretical Computer Science

**Description:** Have you ever written a program with a bug in it? Then this course is for you! Software verification tools can check whether your program works by showing that it correctly satisfies its specification, or finds a case in which it can go wrong. Unlike unit testing and other software validation methods, verification tools use formal methods to rigorously prove correctness. Similar techniques can be used to show that (mathematical models of) cyber-physical systems work as designed.

In this course, we will start by and introducing the main notions of object-oriented program verification, including pre- and post-conditions for methods, and class invariants. We shall use Hoare logic to convert programs and their specifications into logical statements to be proved. We shall apply these techniques to the verification of simple programs written in Java.

In the second part of the course, we consider formal models of software and systems as labelled transition systems (automata), using temporal logics for specification, and consider the fundamental algorithms for verification. We shall apply these algorithms to simple discrete verification problems, such as vending machines and communications systems, modelled using a specification language such as SMV. Finally, we will look at simple continuous systems, such as robots and electronic systems, and show how to verify these using rigorous numerical methods based on interval arithmetic.

**Knowledge and understanding:** By the end of the course, students are able to:

- Recognise the difference between formal verification and validation.
- Explain the various kinds of annotations used in program specification.
- State the deduction and precondition rules of Hoare logic.
- Interpret linear temporal logic formulae.
- Distinguish rigorous numerical methods, notably how interval arithmetic differs from floating-point.

**Applying knowledge and understanding:** Students are able to write formal specifications for simple programs. Furthermore, students can use Hoare logic to reduce program specification to first-order logic statements, and justify these. Students are able to construct Büchi automata accepting temporal logic formulae and can apply interval and affine arithmetic for verifying properties of continuous systems. Moreover, students are able to write annotations for object-oriented software, use software for model-checking discrete systems and use software for rigorous numerics to verify safety of simple continuous systems.

**Making judgements:** Students are able to determine the most appropriate modelling framework and verification tools for a given problem.

**Communication:** Students can write and read formal specifications and can discuss informal design goals and their translation into formal specifications.

**Learning skills:** Students will critically reflect on their own human reasoning and the potential of digital computers.

**Study material:** Course notes.


**Exam:** Written exam (100%)

**ECTS:** 4
Introduction to Quantum Computing (optional course) (Code: KEN3241)

**Examiner:** Dr. G. Stamoulis.

**Prerequisites:** Linear Algebra.

**Desired prior knowledge:** Theoretical Computer Science, Data Structures & Algorithms.

**Description:** This course offers an introduction to the interdisciplinary field of quantum computation. The focus will lie on an accessible introduction to the elementary concepts of quantum mechanics, followed by introducing the mathematical formalism and a comparison between computer science and information science in the quantum domain. The theoretical capability of quantum computers will be illustrated by analysing fundamental algorithms of quantum computation and its potential applications.

Quantum technology has become one of the most prominent interdisciplinary fields of recent research. This course will focus on introducing the mathematical concepts underpinning quantum computation, and on explaining how this new computational paradigm might potentially offer possibilities beyond the scope of conventional computers. Topics that will be introduced and discussed include: (i) most common models of quantum computation (e.g., quantum circuits and measurement-based quantum computing). (ii) An exposition of the machinery borrowed from quantum mechanics, such as superposition of states, quantum entanglement, (de)coherence etc., which gives rise to the potential speed-up of quantum algorithms over their classical analogs. (iii) Some of the most common quantum algorithms (searching, factoring etc.) and protocols (quantum teleportation, EPR paradox). The course will finish with an exposition of potential applications of quantum computation and algorithms in other fields (such as security/cryptography, AI, optimization etc.)

**Important:** no prior knowledge in quantum mechanics is assumed or required, and all necessary concepts will be introduced and motivated from a mathematical and theoretical computer science point of view. Possible quantum architectures and/or related hardware issues will not be discussed.

**Knowledge and understanding:** By the end of this course, students are able to understand the differences between classical and quantum computation: Where is the computational power of quantum machines coming from? What are the limits of this new computational paradigm? What does the term “quantum supremacy” mean and why it is important? How likely is it ever to be achieved and what would it mean for our current understanding of the computational landscape?

**Applying knowledge and understanding:** Students are able to understand some of the most famous quantum algorithms, and to demonstrate where their power comes from. They will be able to judge how this potential computational power can be leveraged, and how it can be applied to other fields in a beneficial way.

After successful completion of this course, students are able to understand and use the mathematical framework of quantum computing to solve computational problems.

**Making judgements:** Students are able to judge and identify the settings where the potential quantum power might be beneficial and how they can leverage this. Students will further be able to analyse simple quantum algorithms for different computational problems.

**Communication:** Students are able to discuss quantum computation critically and judge not only its benefits but, equally important, its shortcomings. During lectures and practical assignments, students will be exposed to a different way of thinking about computation that will also enhance their understanding on classical computation.

**Learning Skills:** Students are able to critically read and understand scientific papers on quantum computing. To explain and analyse quantum algorithms described in quantum circuit or measurement-based quantum computing models. Finally, to relate quantum complexity classes to the classical ones.
Recommended Study material:


Course material will be also provided during the lectures.

Exam: The final course grade is 100% of the final written “closed-book” exam grade.
ECTS: 4

Period 3.3

Project 3-1 (Code: KEN3300)

Examiners: Dr. R. Möckel, Dr. A. Briassouli, Prof. Dr. A. Wilbik, Prof. Dr. N. Tintarev
Tutor(s): Dr. Katharina Schneider, Dr. R. Möckel
Prerequisites: Project 2-1.

Description: Project 3-1 consists of two distinct paths: projects at DKE with focus on university research and DKE/BSSC/BISS projects with focus on applied research proposed by companies that are affiliated with BSSC (Brightlands Smart Service Campus). The DKE/BSSC/BISS projects are facilitated in cooperation with BISS (Brightlands Institute for Smart Society). In the first week of period 1, students indicate their preference by ranking these projects. Groups are created by means of an algorithm that minimizes regret and allocates students to their most preferred options.

About the DKE projects: Students work in small groups, guided by teachers of the subjects concerned and by the tutors. During the project, students apply their knowledge in data science, and artificial intelligence to robotic and other intelligent and autonomous systems. Depending on their chosen specialization within their project group, students study and search for solutions in at least one, typically in multiple, of the following fields: control, machine learning, computer vision, signal processing, human-computer/robot interaction, multi-agent and distributed systems, optimization, data visualization as well as modelling and simulation.

About the DKE/BSSC/BISS projects: Students participate in small groups and receive guidance from a tutor, a teacher with knowledge of the subjects concerned, and a content expert from the company. Furthermore, the students receive business related skills such as creating business presentations from a teacher at BISS. Students learn how to apply their knowledge in data science, and artificial intelligence to solve real-world problem that arise in a professional environment, and how to interact with a client from the industry.

Project Skills period 3.1 & 3.2:

Group CV Check (online or on-site): during this class, you will receive tips and feedback on how to write a professional résumé (i.e. Curriculum Vitae).

Networking skills (online or on-site): during this class, you will learn hands-on tips to build an interesting network to support you in your search for a job or internship.

Both of the above classes are provided by instructors of the UM Career Services.

Study material: Period book 3.1-3.3. Maastricht University, DKE.

Exam: The project will be assessed based on report, product, and presentation.
ECTS: 6
**Period 3.4**

*Data Analysis (Code: KEN3450)*

**Examiner:** Dr. G. Spanakis  
**Tutor:** None.

**Desired Prior Knowledge:** Calculus, Linear Algebra, Mathematical Modelling & Simulation, Machine Learning, Introduction to Computer Science 1 and 2.

**Prerequisites:** Calculus

**Description:** This course aims at preparing students on how to be a successful “data scientist”. The crucial processes of inspecting, cleaning, transforming, restoring and preparing data for modelling are tackled. Different types of data are going to be explored through case studies (“clinics”) that a modern “data scientist” has to deal with. Furthermore, several techniques from machine learning and mathematical modelling (multiple regression, classification, tree-based models, dimensionality reduction, etc.) are presented from the data analysis perspective and students learn how to apply these techniques to different types of data. Finally, the cornerstone of data analysis is presented: correct communication of the analysis outcome (storytelling, visualization, etc.).

**Knowledge and understanding:** Students are able to illustrate and explain data analysis and machine learning techniques with emphasis on modelling, and to give examples of different domains where data analysis can be applied

**Applying Knowledge and understanding:** Students are able to examine datasets using techniques learned in course, and to experiment with different techniques for data modelling

**Making Judgements:** After successful completion of the course, students are able to 1) judge the quality of data (of any kind), 2) to justify and rank which techniques should be applied in each problem and 3) to assess results of data analysis process

**Communication:** Students are able to present the results of different stages of data analysis to specialists and non-specialists and are able to decide on the correct communication medium (scientific, verbal and visual) of the analysis outcome

**Learning Skills:** After successful completion of the course, students are able to suggest options for tackling different datasets combining verbal, numerical/scientific and visual descriptions, also taking into account the context cases (e.g. business, academic) or the domain of application. Furthermore, students are able to formulate data descriptions based on their characteristics and can suggest options for modelling data and perform basic temporal analysis and dimensionality reduction

**Study material:** Jupyter notebooks (and limited slides)  
Recommended literature:

**Selected chapters from the following textbooks:**

A. Downey, Think Stats: Exploratory Data Analysis  
James, G., Witten, D., Hastie, T., Tibshirani: An Introduction to Statistical Learning (with Applications in R)
J. Vanderplans, Data Science Handbook  
S. Skiena, The Data Science Design Manual  
J W. McKinney, Python for Data Analysis  
Chris Albron, Machine Learning with Python Cookbook

**Exam:** Open-book Written Exam 50%, Data clinics 50% (20% individual assignments, 30% group assignments)

**ECTS:** 4
Operations Research Case Studies (Code: KEN3410)

Examiner: Dr. S. Kelk, Dr. S. Chaplick

Desired Prior Knowledge: Linear Programming.

Prerequisites: None.

Description: Operations Research (OR) is concerned with the best way to assign scarce resources to competing activities. It is for this reason an important branch of mathematics that is widely used in industry to support economically efficient decision making, but also in other application areas where discrete or stochastic optimization has a central role. In this course we will explore a number of themes both within deterministic OR (where all the problem data is known at the beginning) and stochastic OR (decision problems involving uncertainty and randomness). Themes within deterministic OR include the network simplex method (used for solving minimum-cost flow problems), integer linear programming and non-linear programming. Stochastic themes include queueing systems, Markov chains and Markov decision problems. As background students will be introduced to the methodological similarities and differences between OR and data science.

Knowledge and understanding: Students can recognize, classify and distinguish some of the major types of OR models, i.e. transportation and network optimization models, integer and non-linear programming, Markov chains and Markov decision problems, queueing models.

Applying knowledge and understanding: Students can apply a wide variety of algorithms to calculate solutions for problems of the types mentioned above. Students will be able to translate simple real-world/industrial optimization problems into a format suitable for (variously) the transportation simplex, network simplex and integer linear programming.

Making judgements: Students can explain advantages and disadvantages of different models and algorithms. They are able to judge the correctness of solutions presented.

Communication: Students can explain and defend their solution methods.

Learning skills: Students will be able to critically reflect upon the scope and limitations of the learned models, and be able to identify follow-up literature describing paradigms, models and algorithms that go beyond the scope of the course.


Recommended literature: None.

Exam: Written exam, worth 100% of the credit.

ECTS: 4

Intelligent Systems (Code: KEN3430)

Examiner: Dr. Ir. K. Driessens

Desired Prior Knowledge: Machine Learning

Prerequisites: None.

Description: The course offers an introduction to intelligent systems, their components, design issues and possible development paths. Based on the metaphor of a computational agent (that is, a software program or a robot which acts and interacts flexibly and autonomously in order to achieve some goal), basic concepts and methods from agent technology are discussed. Topics covered are the concept of artificial intelligence, expert systems, characteristics of an agent and agent architectures, agent cooperation and competition among agents, behaviour-generation and -learning with the added complexity of a multi-agent environment, agent oriented world views and possible future paths to general artificial intelligence. An emphasis is made on the complexity of interacting systems, both between different agents, but also between the subsystems of a single agent. In the practical part of the course, the students build up their experience with the implementation of a number of different types of agents.

Knowledge and understanding: Students are able to compare and discuss benefits and drawbacks of a number of different agent technologies. They can also explain the complexities arising from interactions between multiple techniques within a single agent, and the interactions between agents and systems.
Applying knowledge and understanding: Students will be able to implement a number of different types of agents architectures and agent-subsystems and agent behavior generation techniques.

Making judgements: The student will be able to judge whether it is beneficial to use agent technology over other approaches for handling a given problem, and which agent architecture might fit best.

Communication: The student will gain a working knowledge of agent terminology and will learn to motivate his/her choices concerning the application of agent technology.

Learning Skills: Students have to reflect upon their knowledge and recognize the need for continued learning as they are confronted with the complexities involved with applying the knowledge gained in their bachelor studies and linking individual techniques into a working system.

Study material: Course slides; supplementary material consisting of research papers and book chapters.

Examination: Written exam (80%) + assignments (20%).

ECTS: 4

Period 3.5 to 3.6

Bachelor’s Thesis (Code: KEN3500)
ECTS: 18

Bachelor’s thesis Data Science and Artificial Intelligence
At the end of the Bachelor’s study in Data Science and Artificial Intelligence each individual student has to write a thesis. This thesis has to be designed as a scientific paper of 8 to 10 pages using a standard (LaTeX) design. Students are expected to conduct a pro-active and independent research on their topics. This includes the search and reading of related work. The topics must be discussed with the potential thesis supervisor(s) and a research plan must be submitted to and approved by the Board of Examiners as an initial step. The thesis has to be accompanied by relevant attachments and software. Students will present the thesis in a conference.

This means that a strict submission form will be used. In order to start working on the thesis, a student needs to have obtained at least 140 ECTS (among which are 60 credits of the first year, and 40 ECTS of the second year).

General procedure
Below is an indication for these phases. A special bachelor’s thesis coordinator will supervise the procedure and schedule. Please note there is also an option to start the trajectory in September. For more information ask the thesis coordinator.

November
Phase 0: Thesis Topic meeting
Potential topics and research fields will be presented by staff.

January
Phase 1: Topic selection
During the skills class, each student selects a topic (and problem statement). A thesis supervisor with experience in the field of choice is appointed to each student. Every student hands in a signed bachelor’s project plan to the bachelor’s thesis coordinator. A special bachelor’s thesis coordinator will supervise the phase.

Periods 3.4-3.5: February - May
Phase 2: Research
In this period every student conducts his/her own research. This will preferably be guided in groups by the thesis supervisor. Further two seminars will be organised during which the students present their progress. A special bachelor’s thesis coordinator will supervise the phase.
Phase 3: Writing
Parallel to the research, a scientific article is written.

Period 3.6: June
In Period 6, the research is finished and the first versions of the thesis are expected (first submission). The thesis supervisor will evaluate the paper (or make sure this happens) and gives a first reaction within around a week. A second assessor will also evaluate the paper during this week. The second and final submission will take place at the end of the second week of period 3.6 (concrete dates will be announced).

Phase 4: Preparation for presentation
In the second week of period 3.6, the preparation for the final presentation will start for every student individually. The presentations will be created with PowerPoint and have a maximum length of 10 minutes.

Phase 5: Presentation
The bachelor's theses will be presented in the third week of period 3.6 in a scientific conference setting. The presentations have a maximum length of 15 minutes per student (including questions). The conference is open for all students and teaching staff from Data Science and Artificial Intelligence and anyone else who might be interested. The final decision on the grade for the bachelor's thesis will be made shortly after the presentations. A special bachelor’s thesis coordinator will supervise the phase.

Re-sit: In case the student fails to present his/her work at the Bachelor conference, the student gets one opportunity to defend his/her work at the next bachelor conference. If the student does not participate at or pass any of those two conferences, the student has to select a new topic and submit a new thesis plan. For students not finishing at the June Conference there is one re-sit possibility in a Conference at the end of August.

Requirements for the bachelor’s thesis project
For the bachelor’s thesis, every student has to conduct a short scientific research project. This can be empirical as well as a theoretical research. The topic for the research project is open, as long as it fits into the Data Science and Artificial Intelligence program. The department will offer a list of potential research topics. The topic and the research questions have to be approved by the thesis supervisor. To achieve this, the student will create a bachelor’s project plan using the form provided by the Board of Examiners which contains the following:
• Date;
• Name of the thesis supervisor;
• Name of the second examiner;
• Title of the bachelor’s thesis, start- and end date of the thesis project and a planning;
• Short description of the research question;
This plan will be signed by the student, the thesis supervisor and the second examiner and then handed in to the bachelor’s thesis coordinator. It is possible to execute the bachelor thesis project as an external training period. This should be well defined in the bachelor's thesis plan. In this case, the plan should also include the name of the company, the name of the external supervisor, the size of the project and any agreements about compensation and confidentiality. The plan should also be signed by the external supervisor. The external research cannot start before period 3.5. The research needs to be original in such a way that the thesis supervisor is convinced that this research has not been done before. The research also needs enough depth and still it must be possible to finish it in the set amount of time. It is possible for multiple students to cooperate in a research project as long as it is clear who did what. Moreover, every student has to write his own thesis reflecting their part of the research.
Requirements for the bachelor’s thesis document

Content aspects
The thesis describes the cause, research question, approach and results of the research. This has to be done in a clear, structured and scientific manner which includes:

- a clear introduction in which the cause and research questions are presented;
- a clear conclusion, based solely on the already used thought out principles and derived results;
- a clear line is shown between problem statements approach methods and the derived results;
- a motivation of the followed approach;
- an adequate description of the followed approach;
- an honest, clear and concise description of the derived results, if necessary using tables;
- a discussion of the results;
- the usage of relevant and recent literature;
- the correct usage of references;
- the adequate usage of the literature for the reasoning in the thesis.

Design aspects
The number of pages of the thesis is between 8 and 10 (no more, no less) in the designated LaTeX format, including images and references. This thesis should at least contain:

- title;
- author;
- abstract;
- one or two keywords;
- list of references;
- page numbers.

It goes without saying that the correct, scientific references are used for used resources (by using the designated BiBTeX reference style). Images and tables are accompanied by an index and caption. Mathematical formula, definitions, etc. have to be properly designed and numbered. The start and end of mathematical formula have to be properly defined.

Language aspects
The thesis has to be written in Dutch or English, considering correct spelling, syntactical structure of sentences and structure of content in paragraphs. The target audience is fellow Data Science & Artificial Intelligence students. Any jargon and/or abbreviations have to be explained unless they are common knowledge for this audience (e.g. CPU).

Citations
It is allowed to use several short citations with a maximum length of two sentences. These citations have to be clearly referenced and have to be typographically distinguishable (that is, citations are placed in quotes). Non-allowed citations or missing references will result in an unsuccessful result.

Assessment
The assessment will take place based on the contents and design of the thesis and the presentation of this thesis.
2.4 DKE Honours programme

DKE offers its talented and top-performing bachelor’s students the possibility to participate in the DKE Honours Programme. This programme offers two variants MaRBle 2.0 and KE@Work.

MaRBle 2.0
In MaRBle 2.0, you will get the opportunity to work on a state-of-the-art research project. Work will be organized in a similar way as in professional research institutes where participants work together as individual experts on a team project. For more information on the MaRBle 2.0 please contact Dr. Mathias Staudigl (m.staudigl@maastrichtuniversity.nl). MaRBle takes place in year two of the bachelor’s programme.

KnowledgeEngineering@Work (KE@Work)
Students admitted to the KE@Work path are placed at a company or organisation through a careful selection and matching process. During the full second and third year of the bachelor program, they spend 50% of the time in class and 50% at the company, where they work on solving academic challenges and complex business problems, under supervision of dedicated business and university supervisors. For more information on KE@Work, please contact our colleagues via kework@maastrichtuniversity.nl

Selection of honours students will happen in the second semester of year 1. If you successfully complete the honours programme, this will be certified on an honour’s diploma supplement.
3 Master

3.1 Curriculum of the First Year of the Master’s Programmes

General introduction Master AI
The impact of the field of Artificial Intelligence is pertinent due to the key role it plays in technological applications that have become indispensable in society, such as simple personal assistants that adapt the settings of your smart phone to automatically recognised activities (e.g. driving or attending a meeting, automated trading software used in real markets to respond to rapid price changes, interactive computer games that include human like opponents, robotic assistance in the exploration of dangerous environments, etc.). In this master’s programme, you are trained to become an expert and capable of dealing with today’s and future challenges in the field of Artificial Intelligence.

The master’s programme Artificial Intelligence covers a range of subjects emphasizing the following research topics as its core:
1) Intelligent techniques for playing and solving (board) games and controlling virtual characters in video games;
2) Situated agents to study the control and coordination of embodied agents, i.e. robots (e.g. autonomous flying robot swarms);
3) Multi-agent systems of collaborating autonomous intelligent systems;
4) Formal techniques for reasoning in agents and representing and communicating knowledge;
5) Machine learning to extract useful patterns and knowledge from experience and make predictions about the future;

The members of the teaching staff are actively involved in one or more of these research topics. As a result, the educational contents of the courses relate directly to the research performed.
## Programme master's AI Year 1

<table>
<thead>
<tr>
<th>Period</th>
<th>Course Name</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1</td>
<td>Foundations of Agents (KEN4115)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Intelligent Search &amp; Games (KEN4123)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Research Project 1 (**)</td>
<td>6</td>
</tr>
<tr>
<td>Period 2</td>
<td>Multi-Agent Systems (KEN4111)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Advanced Concepts in Machine Learning (KEN4154)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Research Project 1 (**)</td>
<td>6</td>
</tr>
<tr>
<td>Period 3</td>
<td>Research Project 1 (KEN4130)</td>
<td>6</td>
</tr>
<tr>
<td>Period 4</td>
<td>Autonomous Robotic Systems (KEN4114)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1 Elective Course from the following set:</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Algorithms for Big Data (KEN4254)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dynamic Game Theory (KEN4251)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computational Statistics (KEN4258)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Advanced Natural Language Processing (KEN4259)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research Project 2 (**)</td>
<td></td>
</tr>
<tr>
<td>Period 5</td>
<td>2 Elective Courses from the following set:</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Computer Vision (KEN4255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Information Retrieval and Text Mining (KEN4153)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deep Learning (KEN4257)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Planning and Scheduling (KEN4253)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Explainable AI (KEN4246)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research Project 2 (**)</td>
<td></td>
</tr>
<tr>
<td>Period 6</td>
<td>Research Project 2 (KEN4131)</td>
<td>6</td>
</tr>
</tbody>
</table>

### Programme master's AI Year 2

<table>
<thead>
<tr>
<th>Period</th>
<th>Course Name</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1, 2, 3</td>
<td>Electives *</td>
<td>30</td>
</tr>
<tr>
<td>Period 4, 5, 6</td>
<td>Master's thesis AI (KEN4160)</td>
<td>30</td>
</tr>
</tbody>
</table>

* Note: during the elective semester (first semester of year 2) of the master's programme it is possible to take electives from our other master's programme or relevant master's programmes at Maastricht University (maximum of 13 ECTS outside DKE) or to participate in a research project, a company internship or a study abroad semester at one of our partner universities. Please contact exchange officer and/or the Study Adviser for more information.

** The Research Project 1 will start in period 1.1 and runs until the end of period 1.3 with weekly meetings. The credits for the project will become available at the end of period 1.3. The Research Project 2 will start in period 1.4 and runs until the end of period 1.6 with weekly meetings. The credits for the project will become available at the end of period 1.6.
Master's Data Science for Decision Making

General introduction Master DSDM
Data Science for Decision Making is the science of making informed decisions. It has widespread applications in business and engineering. In today's world, many companies and organisations collect all sorts of data in large amounts. They aim to extract useful information from it, to recognize patterns and anomalies. Data Science for Decision Making provides the mathematical tools to analyse and model these big data. It also provides and uses the computational software that is the key to data science.

The two-year master's programme in Data Science for Decision Making teaches the use of applied mathematics to analyse and optimize processes, problems and operations. Examples of applications are: discovering patterns in data such as images and time series, scheduling customer service agents, optimising supply chains, controlling dynamical systems, modelling biological processes, finding optimal strategies in negotiation, and extracting meaningful components from brain signals.

The master’s programme Data Science for Decision Making covers a wide range of research topics, focusing on the following ones in its core:
1) Data mining to extract useful patterns and knowledge from large data repositories;
2) Mathematical modelling and parameter estimation from data, system identification, model approximation and reduction of model complexity;
3) Algorithm design and analysis to efficiently deal with the challenges that the ever-growing amount of data pose;
4) Mathematics and algorithms associated with modelling and solving planning/scheduling problems.

The members of the teaching staff are actively involved in one or more of the research topics. As a result, the educational contents of the courses relate directly to the research performed.
<table>
<thead>
<tr>
<th>Period</th>
<th>Course/Project</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1</td>
<td>Data Mining (KEN4113)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1 Elective Course from the following set:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Signal and Image Processing (KEN4222)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mathematical Optimization (KEN4211)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stochastic Decision-Making (KEN4221)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research Project 1 (**)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Research Project 1 (**)</td>
<td></td>
</tr>
<tr>
<td>Period 2</td>
<td>Model Identification and Data Fitting (KEN4242)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1 Elective Course from the following set:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Advanced Concepts in Machine Learning (KEN4154)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Applications of Image and Video Processing (KEN4244)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research Project 1 (**)</td>
<td>6</td>
</tr>
<tr>
<td>Period 3</td>
<td>Research Project 1 (KEN4130)</td>
<td>6</td>
</tr>
<tr>
<td>Period 4</td>
<td>Algorithms for Big Data (KEN4254)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1 Elective Course from the following set:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Building and Mining Knowledge Graphs (KEN4256)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Dynamic Game Theory (KEN4251)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computational Statistics (KEN4258)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Advanced Natural Language Processing (KEN4259)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Fusion (KEN4223)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research Project 2 (**)</td>
<td>6</td>
</tr>
<tr>
<td>Period 5</td>
<td>Planning and Scheduling (KEN4253)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1 Elective Course from the following set:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Symbolic Computation and Control (KEN4252)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Deep Learning (KEN4257)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Information Retrieval and Text Mining (KEN4153)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computer Vision (KEN4255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Algorithms for Data Visualization (KEN4213)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Explainable AI (KEN4246)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research Project 2 (**)</td>
<td>6</td>
</tr>
<tr>
<td>Period 6</td>
<td>Research Project 2 (KEN4131)</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Programme master's DSDM Year 2</th>
<th>ECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1, 2, 3</td>
<td>Electives *</td>
</tr>
<tr>
<td>Period 4, 5, 6</td>
<td>Master's thesis DSDM (KEN4260)</td>
</tr>
</tbody>
</table>

* Note: during the elective semester (first semester of year 2) of the master’s programme it is possible to take electives from our other master’s programme or relevant master’s programmes at Maastricht University (maximum of 13 ECTS outside DKE) or to participate in a research project, a company internship or a study abroad semester at one of our partner universities. Please contact exchange officer and/or the Study Adviser for more information.

** The Research Project 1 will start in period 1.1 and runs until the end of period 1.3 with weekly meetings. The credits for the project will become available at the end of period 1.3. The Research Project 2 will start in period 1.4 and runs until the end of period 1.6 with weekly meetings. The credits for the project will become available at the end of period 1.6.
Part of the Research project is a Project skill programme, offered in September and February (all elements are mandatory):

Period 1.1:
- Session 1 (1 hour): Introduction to Project-Centred Learning and LateX
- Session 2 (3 hours): Team Building afternoon
- Session 3 (2 hours): Discover your Competences Part 1 (Career Services)
- Session 4 (2 hours): Discover your Competences Part 2 (Career Services)
- Session 5 (2 hours): Presentation skills
- Session 6 (2 hours): Presentation skills

Period 1.2:
- Session 1 (3 hours): Academic Writing Skills
- Session 2 (3 hours): A workshop of choice at Career Services

Note: Master’s students who have finished their bachelor’s degree at the DKE and already participated in the introduction to PCL and Latex do not need to partake in session 1, period 1.1. Moreover, if these students can show that they already took the workshop “discover your competences” they may request for an exemption with the study advisor.

Period 1.1

Foundations of Agents (Code: KEN4115)

Examiner: Dr. Ir. Ing. N. Roos.


Prerequisites: A basic course in logic and in probability theory.

Description: Agents are autonomous computer programs, robots, humans, etc. Agents operate in some environment, which they can observe, and in which they can realize objectives through the execution of actions. Examples of environment in which agents can operate, are computer game environments, the internet, and also the physical world is case of robots and humans.

In this course we address the problem of how an agent can act optimally in order to realize its objectives. We will answer this question by investigating how we can formally specify the agent’s environment, the agent’s objectives, the observations the agent can make and the actions it can execute. We use the formal model to investigate how the agent can determine an (optimal) behaviour realizing its objectives.

The following formal models will be investigated:
- Markov Decision Processes,
- Partially Observable Markov Decision Process,
- logic-based models such as Epistemic Logic, Doxastic Logic, Dynamic Logic, and BDI logics, and
- Game Theory.

Some examples of methods for determining the agents optimal behaviour addressed in the course are: Value and Policy Iteration, Q-Learning, Planning, etc.
Knowledge and understanding:
- The student will be able to explain formal models for describing agents.
- The student will be able to verify whether the formal models are correct.
- The student will be able to explain the underlying assumptions of each formal model.
- The student will be able to compare and discuss the differences between the formal models.
- The student will be able to analyze important properties of formal models for describing agent.

Applying knowledge and understanding:
- The student will be able to apply formal models for describing agents to solve practical problems.
- The student will be able to assess which formal model is most suited for addressing a practical problem.
- The student will be able to make a practical implementation of a formal model.

Making judgments:
- The student will be able to analyze which formal model of an agent is adequate for specific problem domains.
- The student will be able to select the formal model of an agent that is adequate for specific problem domains.

Communication:
- The student will be able to explain to its peers why a formal model of an agent is adequate for specific problem domains.
- The student will be able to explain to its peers how a formal model of an agent should be applied for a specific problem.

Learning skills:
- The student will be able to explain to its peers why a formal model of an agent is adequate for specific problem domains.
- The student will be able to explain to its peers how a formal model of an agent should be applied for a specific problem.

Study material: Syllabi, scientific papers.
Recommended Literature: none.
Examination: Written exam at the end of the course. A bonus of 1.0 point can be earned by a series of bonus assignments.
ECTS: 6
Intelligent Search & Games (Code: KEN4123)

Examiners: Prof. dr. M.H.M. Winands and dr. C. Browne.

Desired Prior Knowledge: Data Structures & Algorithms

Course description: In this course, the students learn how to apply advanced techniques in the framework of game-playing programs. Depending on the nature of the game, these techniques can be of a more or less algorithmic nature. The following subjects will be discussed:

1. Basic search techniques. Alpha-beta; A*.
2. Advanced search techniques. IDA*; B*, transposition tables; retrograde analysis and endgame databases; proof-number search and variants; multi-player search methods; Expectimax and *-minimax variants.
3. Heuristics. World representations; killer moves; history heuristic, PVS; windowing techniques; null-moves; forward-pruning techniques; selective search, GOAP.
4. Monte Carlo methods. Monte Carlo tree search (MCTS) techniques, enhancements and applications; AlphaGo and AlphaZero approaches.
5. Game design. Evolutionary game design; game quality metrics; self-play evaluation; procedural content generation (PCG); puzzle design.

Knowledge and understanding: The student can explain basic and advanced search techniques and can identify which of them to use either in a game context, or in problems with a similar structure. Applying knowledge and understanding: Students have obtained the knowledge to develop, program, analyse, and apply advanced techniques autonomously to a wide variety of problems. They will also learn that adapting known techniques to fit a given problem can achieve a better performance.

Making judgements: Students will be able to judge the quality of approaches (systems or scientific publications) based on the techniques taught.

Communication: Students will be able to present the results of their game programs and search algorithms to specialists or non-specialists.

Learning skills: Students will be able to familiarize themselves with Game AI techniques beyond the scope of the course in order to solve a problem.

Study material: Course notes and other information made available.

Recommended Literature:

Exam: Written exam (50%) + a large practical task (50%).

ECTS: 6
Data Mining (Code: KEN4113)

Examiners: Dr. E. N. Smirnov

Desired Prior Knowledge: Statistics and Basic Machine Learning

Prerequisites: None.

Course description: Data mining is a major frontier field of computer science. It allows extracting useful and interesting patterns and knowledge from large data repositories such as databases and the Web. Data mining integrates techniques from the fields of databases, machine learning, statistics, and artificial intelligence. This course will present the state-of-the-art techniques of data mining. The lectures and labs will emphasize the practical use of the presented techniques and the problems of developing real data-mining applications. A step-by-step introduction to data-mining environments will enable the students to achieve specific skills, autonomy, and hands-on experience. A number of real data sets will be analysed and discussed.

Knowledge and understanding: Students will acquire knowledge on data preparation, data preprocessing, feature selection/generation, data mining, and model validation.

Applying knowledge and understanding: When confronted with real-life problems, students will be able to identify data-analysis tasks. Then, they will be able to apply data-mining techniques for supervised and unsupervised data-analysis. If necessary, students will be able to design data-mining algorithms specific for the tasks they have.

Making judgements: Students will be able to assess the quality of data-mining models, processes, results, and tools.

Communication: Students will be able to present the results of different stages of data-mining processes to specialists or non-specialists.

Learning skills: Students will be able to recognize their own lack of knowledge and understanding and take appropriate action such as consulting additional material or other sources of help.

Study material: Course notes, slides, and other information made available.

Recommended Literature:

Exam: Written exam (80%) + practical assignments (20%).

ECTS: 6
Signal and Image Processing (Code: KEN4222)  

**Examiner:** Dr J. Karel and dr. P. Bonizzi  

**Desired Prior Knowledge:** Linear algebra, Calculus, basic knowledge of Matlab. Some familiarity with linear systems theory and transforms (such as Fourier and Laplace) is helpful. 

**Prerequisites:** None.  

**Course description:** This course offers the student a hands-on introduction into the area of digital signal and image processing. We start with the fundamental concepts and mathematical foundation. This includes a brief review of Fourier analysis, z-transforms and digital filters. Classical filtering from a linear systems perspective is discussed. Next wavelet transforms and principal component analysis are introduced. Wavelets are used to deal with morphological structures in signals. Principal component analysis is used to extract information from high-dimensional datasets. We then discuss Hilbert-Huang Transform to perform detailed time-frequency analysis of signals. Attention is given to a variety of objectives, such as detection, noise removal, compression, prediction, reconstruction and feature extraction. We discuss a few cases from biomedical engineering, for instance involving ECG and EEG signals. The techniques are explained for both 1D and 2D (images) signal processing. The subject matter is clarified through exercises and examples involving various applications. In the practical classes, students will apply the techniques discussed in the lectures using the software package Matlab. 

**Knowledge and understanding:** Students are able to explain fundamental concepts of signal and image processing and their mathematical foundation. They are able to 1) describe various types of filters and their properties, 2) explain orthogonal wavelet filter banks and describe their properties, 3) explain a construction scheme and elicit a wavelet-based noise-filtering scheme, 4) explain principal component analysis and empirical signal processing techniques and how they complement the other techniques discussed. 

**Applying knowledge and understanding:** Students are able to use the various techniques discussed during the lectures to solve real-world problems, such as being able to apply wavelet filtering and principal component analysis on various signals. They are also able to analyse a signal by using Matlab, and independently interpret the outcome of an analysis. 

**Making judgements:** Students are able to assess what technique is suited for a signal processing problem at hand, and to independently and critically look at a signal or image, and understand if and what type of pre-processing is required. 

**Communication:** Students are able to communicate signal and image processing techniques and strategies, and the results of their analyses to experts and non-experts. 

**Learning skills:** Students are able to independently master signal and image processing techniques, from classical signal processing techniques to more empirical techniques. 


Additional material provided electronically on Student Portal. 


**Exam:** Written exam/Computer exam. 

**ECTS:** 6
Mathematical Optimization (Code KEN4211)

Examiner: Dr. P. Collins

Desired prior knowledge: Simplex algorithm. Calculus, Linear Algebra.

Description: Optimization (or “Optimisation”) is the subject of finding the best or optimal solution to a problem from a set of potential or feasible solutions. Optimization problems are fundamental in all forms of decision-making, since one wishes to make the best decision in any context, and in the analysis of data, where one wishes to find the best model describing experimental data. This course treats two different areas of optimization: nonlinear optimization and combinatorial optimization. Nonlinear optimization deals with the situation that there is a continuum of available solutions. A best solution is then usually approximated with one of several available general-purpose algorithms, such as Brent’s method for one-dimensional problems, Newton, quasi-Newton and conjugate gradient methods for unconstrained problems, and Lagrangian methods, including active-set methods, sequential quadratic programming and interior-point methods for general constrained problems. Combinatorial optimization deals with situations that a best solution from a finite number of available solutions must be chosen. A variety of techniques, such as linear programming, branch and cut, Lagrange relaxation dynamic programming and approximation algorithms are employed to tackle this type of problems. Throughout the course, we aim to provide a coherent framework for the subject, with a focus on consideration of optimality conditions (notably the Karush-Kuhn-Tucker conditions), Lagrange multipliers and duality, relaxation and approximate problems, and on convergence rates and computational complexity. The methods will be illustrated by in-class computer demonstrations, exercises illustrating the main concepts and algorithms, and modelling and computational work on case studies of practical interest, such as optimal control and network flow.

Knowledge and understanding: By the end of this course, students will have a strong foundation in nonlinear and combinatorial optimization. You will be able to formulate real-life problems as optimization problems. You will understand optimality conditions, including the Karush-Kuhn-Tucker conditions and be able to test for optimality. You will know how to solve a variety of general optimization problems, including constrained nonlinear problems, and (mixed-) integer linear problems. You will understand notions of duality and Lagrange multipliers, and be able to apply techniques based on relaxation and approximation.

Applying knowledge and understanding: Students will know the advantages and disadvantages of different methods, and be able to choose an appropriate method for a given problem. You will be able to implement and test optimization algorithms on a computer. You will be able to apply your knowledge to the solution of practical problems and in developing new efficient algorithms.

Making judgements: Students will be able to select an appropriate solution method for a given optimization problem, and judge the quality of the solution obtained.

Communication: Students will be able to discuss the development and use of optimization algorithms.

Learning skills: Students will learn how to develop and implement mathematical methods, select and evaluate algorithms, and formulate mathematical model of real-world problems.

Study material: Lecture notes, handouts.

Recommended literature:
1. Numerical Optimization, by Nocedal and Wright (Springer)
2a. Combinatorial Optimization, Algorithm and Complexity, by Papadimitriou and Steiglitz (Dover Publications), or

Exam: Written exam, closed book with formula sheet (100%)

ECTS: 6
Stochastic Decision Making (Code: KEN4221)

Examiner: Dr. G. Schoenmakers

Prerequisites: Probability & Statistics.

Course Description: Any realistic model of a real-world phenomenon must take into account the possibility of randomness. That is, more often than not, the quantities we are interested in will not be predictable in advance but, rather, will exhibit an inherent variation that should be taken into account by the model. Mathematically, this is usually accomplished by allowing the model to be probabilistic in nature. In this course, the following topics will be discussed:

(1) Basic concepts of probability theory: Probabilities, conditional probabilities, random variables, probability distribution functions, density functions, expectations and variances.

(2) Finding probabilities, expectations and variances of random variables in complex probabilistic experiments.

(3) Discrete and continuous time Markov chains and related stochastic processes like random walks, branching processes, Poisson processes, birth and death processes, queueing theory.

(4) Markov decision problems.

Knowledge and understanding: In this course, the students acquire tools for modelling complex processes involving randomness, providing a basis for originality in developing and/or applying ideas in a research context.

Applying knowledge and understanding: When confronted with complex problems that involve probabilistic experiments, students have the tools to create and analyse appropriate models.

Making judgements: The students are able to analyse complex problems as stochastic processes and solve them. Furthermore, students can find optimal solutions in decision problems that are based on these stochastic processes.

Communication: The students will be able to communicate their conclusions and the underlying rationale to expert and non-expert audiences.

Learning Skills: The students have obtained the skills to study related material in a largely autonomous manner.

Study material: Introduction to Probability Models by Sheldon M. Ross (9th or 10th ed.) + Lecture notes that are provided via Student Portal.

Recommended Literature: Probability: A Lively Introduction by Henk Tijms.

Exam: Written exam.

ECTS: 6
Multi-Agent Systems (Code: KEN4111)

Examiner: Prof. dr. G. Weiss.

Desired Prior Knowledge: Introduction to Computer Science 1 and 2.

Description: Multi-agent systems are systems composed of multiple interacting intelligent agents, where an agent is a computational entity such as a software program or a robot that is situated in some environment and that to some extent is able to act autonomously in order to achieve its design objectives. The field of multi-agent systems has its origin in the late 1970s and today is an established and vibrant topic in computer science. Multi-agent systems are an enabling technology for applications that rely on distributed and parallel processing of data, information and knowledge in complex – networked, open and large-scale – computing environments. With advancing technological progress in inter-connectivity and interoperability of computers and software such applications are becoming standard in a variety of domains such as e-commerce, logistics, supply chain management, telecommunication, health care, and manufacturing. The course covers important conceptual, theoretical and practical foundations of multi-agent systems. Examples of topics treated in the course are agent-agent communication, automated negotiation and argumentation in cooperative and competitive settings, multi-agent learning and planning, automated decision making based on mechanisms such as voting and auctioning, and development and engineering of agent-based systems. In the practical part of the course students have the opportunity to apply the learnt multi-agent concepts, algorithms and methods.

Knowledge and understanding: The student is able to describe existing and design novel agent-agent coordination mechanisms and interaction principles, and can explain and analyse their strengths and shortcomings.

Applying Knowledge and Understanding: The student is be able to apply the gained knowledge in concrete application scenarios and practical applications.

Making Judgements: The student is be able to judge for a given problem whether and in how far it is beneficial to use a multi-agent approach for its solution.

Communication: The student is able to motivate and explain benefits and shortcomings of their usage in a given application, and thereby showing sufficient understanding of multi-agent concepts.

Learning Skills: The student is able to study autonomously multi-agent system literature, including, in particular, literature describing new developments in the methods and techniques covered in this course.

Study material: Course slides; supplementary material to be announced.

Recommended literature:


Examination: Practical and reading assignments (30%) and written exam (70%)

ECTS: 6
Advanced Concepts in Machine Learning (Code: KEN4154)

**Examiner:** Dr. ir. K. Driessens  
**Desired Prior Knowledge:** Machine Learning  
**Prerequisites:** None.  
**Description:** This course will introduce a number of advanced concepts in the field of machine learning such as Support Vector Machines, Gaussian Processes, Deep Neural Networks, etc. All of these are approached from the view that the right data representation is imperative for machine learning solutions. Additionally, different knowledge representation formats used in machine learning are introduced. This course counts on the fact that basics of machine learning were introduced in other courses so that it can focus on more recent developments and state of the art in machine learning research. Labs and assignments will give the students the opportunity to implement or work with these techniques and will require them to read and understand published scientific papers from recent Machine Learning conferences.

**Knowledge and understanding:** Students can explain, construct and adapt powerful machine learning techniques, most with a statistical background. Students recognise the need for non-standard techniques and representations that can be used for complex/structured data. They can explain the strengths and weaknesses of different machine learning approaches.

**Applying knowledge and understanding:** Students will be able to select, adapt and apply a number of advanced machine learning approaches. They will be able to select the correct representation for a machine-learning problem and to translate a machine learning problem into a suited representation format.

**Making judgements:** Students will be able to judge which machine learning approach and data-representation is best suited. They will also be able to comprehend and judge machine-learning research.

**Communication:** Students will be able to relate different machine learning techniques to each other and explain their working, benefits and disadvantages to non-experts. They will also be able to discuss the need and use of structured representation with both experts and non-experts.

**Learning Skills:** Students will be able to relate information from different sources, and read process and evaluate recent research developments in the field of machine learning.

**Study material:** Slides and collected notes and chapters from freely available books and course notes.

**Recommended literature:** Pattern Recognition and Machine Learning - C.M. Bishop; Bayesian Reasoning and Machine Learning - D. Barber; Gaussian Processes for Machine Learning - C.E. Rasmussen & C. Williams; The Elements of Statistical Learning - T. Hastie et al.

**Exam:** Students are graded using a number of assignments (20%), a report on a recent scientific paper or a machine learning application (10%) and a written exam (70%).

**ECTS:** 6
Model Identification and Data Fitting (Code: KEN4242)


Tutor(s): None.

Desired Prior Knowledge: Basic knowledge of Matlab and some familiarity with linear systems theory and transforms (such as Fourier and Laplace) is helpful. This course offers a useful prior knowledge for the course Symbolic Computation and Control.

Prerequisites: Linear Algebra, Mathematical Modelling, Probability and Statistics.

Course description: This course is devoted to the various practical and theoretical aspects which involve the estimation (the identification) of a mathematical model within a given model class, starting from a record of observed measurement data (input-output data). First, we address distance measures, norms, and criterion functions. Then we discuss the prediction error identification of linear regression models, with special emphasis on the various interpretations of such models (deterministic, stochastic with Gaussian white noise and maximum likelihood estimation, stochastic in a Bayesian estimation context) and on numerical implementation aspects (recursion, numerical complexity, numerical conditioning and square root filtering). Next, we study identification within the important class of auto-regressive dynamical models, to which the Levinson algorithm applies.

Other related topics receiving attention are identifiability, model reduction and model approximation. Some techniques for the estimation of linear dynamical i/o-systems are illustrated with the system identification toolbox in Matlab.

Knowledge and understanding: Students learn to recognize the various aspects that play a key role in building a mathematical model from measurement data: the choice of model class (and order), the choice of parameterization, the criterion of fit, the model estimation method, the quality of the measurement data, and the validity of the estimated model.

Applying knowledge and understanding: Students are able to 1) estimate models from measurement data, particularly linear regression models and auto-regressive models, 2) to assess the quality of a (linear regression) model, and 3) assess whether a model is identifiable.

Making judgements: Students are able to predict and judge the quality of models that can be obtained from a record of measurement data.

Communication: Students learn to motivate the choice of a model class, the model order and an estimation method to identify a model from measurement data, to interpret the identification outcomes and to explain all this to specialists and non-specialists.

Learning skills: Students are able to read and interpret scientific literature on model estimation and system identification, and to use Matlab and work out ideas computationally.

Study material: Syllabus, provided electronically on the digital learning environment.

Recommended literature:

Exam: Written exam.

ECTS: 6
Applications of Image and Video Processing (KEN4244)

Examiner: Dr. A. Briassouli


Prerequisites: None.

Description: Applications of image and video processing will be presented, and connections to basic algorithms will be demonstrated. We will examine some of the most popular and widespread applications, namely security, surveillance, medical, traffic monitoring, astronomy, farming, culture. The methods used in these applications will be analysed in class and common characteristics between them will be explained. Students will be able to suggest further applications of interest to them and bring relevant literature to the class.

Knowledge and understanding: Students will acquire a wide-ranging understanding of the latest trends in image and video processing methods and how these are applied in real world applications. They will obtain insights on common problems encountered in these applications, and how they can be tackled through advanced image and video processing algorithms.

Applying knowledge and understanding: The knowledge and understanding obtained in this class will be demonstrated in mini projects based on State of the Art research.

Making judgements: Through the presentation of various applications of image and video processing, students will be able to analyse problems in the real world, and understand how to best address them.

Communication: Part of the class will include homework where students will carry out a short literature review and implementation of mini projects on applications that interest them. They will be taught how to communicate them succinctly and effectively, maintaining a balance between overall understanding and technical depth.

Learning Skills: Students will obtain a spherical comprehension of connections between machine learning and image/video/signal processing, as well as their practical in a wide range of applications in our daily life. They will be able to identify the methods needed in different applications of image and video processing, propose a plan for solving the corresponding problems, and justify it.

Study material: Lecture slides, selected papers.

A Bovik, Handbook of Image and Video processing

Recommended literature:

Exam: Mini projects (50%) and final exam (50%)

ECTS: 6
Period 1.3

Research Project AI 1 & DSDM 1 (Code KEN4130 & KEN4230)

Examiner: T.b.a.
Coordinator: Dr. ir. Kurt Driessens
Desired Prior Knowledge: None.
Prerequisites: None.
Description: The research project takes place during the three periods of the semester. Project topics are presented at the start of the semester and assigned to students based on their preferences and availability. The emphasis in the first phase is on initial study of the context set out for the project and the development of a project plan. In the second period, the goal is to start modelling, prototyping and developing. At the end of period 1 and 2, a progress presentation takes place. In period 3, the implementation, model and/or experiments set out in the project plan has to be finished and reported on. The project results in a project presentation, a project report and possibly a public website and/or product.

Knowledge and understanding: Students get to know and possibly contribute to state of the art methods within the fields of Artificial Intelligence and/or Data Science for Decision Making to answer an open question.

Applying knowledge and understanding: Student write their own research plan in coordination with a staff member (plus possibly outsiders) who act as clients with an open question. Students with different backgrounds and from both masters work together in teams to build and evaluate an answer to an open question. Students find, judge the suitability, apply, and evaluate state of the art techniques to answer questions and construct applications in the field of Artificial Intelligence and Data Science. Students apply the accumulated knowledge from other educational activities in application specific areas.

Making judgements: Students judge feasibility of tasks, attainability of goals, and the amount of work involved. Students think about the possible consequences of their work. Students evaluate state of the art and the applicability and scope of research results.

Communication: Students will learn to:

1. orally communicate and cooperate with peers
2. orally report on progress and intermediate results to superiors
3. orally negotiate and communicate with clients
4. communicate their ideas in written form, both for an academic and a general audience
5. give effective presentations

Learning Skills: Students increase their own level of knowledge in a specialised sub-discipline of the field of Artificial Intelligence and/or Data Science. Students perform research into recent state of the art techniques. Students learn that the field of Artificial Intelligence and Data Science are constantly evolving beyond what is taught in class.

Study material: Slides provided at the end of joint information sessions. Literature provided by the project supervisors.


Exam: Phase 1: project plan + presentation (15%); Phase 2: layman’s website + presentation (15%); Phase 3: Project report + presentation (70%)

ECTS: 6
Autonomous Robotics Systems (Code: KEN4114)

Examiner: Dr. R. Möckel.


Prerequisites: None

Description: Operating autonomously in unknown and dynamically changing environments is a core challenge that all robotic systems must solve to work successfully in industrial, public, and private areas. Currently popular systems that must demonstrate such capabilities include self-driving cars, autonomously operating drones, and personal robotic assistants. In this course, students obtain deep knowledge in creating autonomous robotic systems that can operate in and manipulate unknown and dynamically changing environments by autonomously planning, analysing, mapping, and modelling of such environments. Students learn to approach these challenging tasks through three main techniques: swarm intelligence, model-based probabilistic frameworks, and (mostly) model-free techniques from artificial evolution and machine learning.

Knowledge and understanding: Students gain a deep understanding of the challenges in autonomous robotic systems and how these challenges are addressed in state-of-the-art systems. Students learn about and practise techniques for autonomous mapping, localization, navigation, sensing, modelling robot motion, planning, and decision-making. Through the course, students obtain in-depth knowledge and hands-on experience in a variety of algorithms and techniques from machine learning, agent technology, and search techniques including Bayesian filters (like Kalman Filters, Extended Kalman Filters, Histogram Filters, and Particle Filters), artificial neural networks, evolutionary algorithms, and swarm intelligence.

Applying knowledge and understanding: After successful completion of the course, students will have obtained in-depth knowledge to understand, adapt, apply, and autonomous robotics systems. Students obtain the ability to select from a variety of available tools feasible solutions for the complex and rather ill defined problem domains of autonomous robotic systems and to predict the resulting consequences of their choices. Furthermore, students learn how to choose, apply, formulate, and validate models of autonomous robotic systems and of appropriate control techniques from artificial intelligence for these systems.

Making judgements: Students will be able to comprehend and to critically judge scientific publications on autonomous systems, artificial evolution, and swarm intelligence. From this literature, students are able to search for and to critically process information to solve given ill-defined but in practise highly relevant problems in autonomous systems. Students are able to critically discuss social, economic, and ethical consequences of artificial intelligence and autonomous decision-making.

Communication: Students learn to critically discuss challenges and professional solutions in autonomous robotic applications with both experts and non-experts.

Learning Skills: The course prepares students to work on robotic applications in professional research and business environments. Students will be able to autonomously acquire new skills and knowledge to develop, program, analyse, and apply advanced techniques to a wide variety of problems.


Recommended literature:

Examination: The final course grade is 80% of the final written “closed-book” exam grade plus 20% of the practical group assignments grades

ECTS: 6
Algorithms for Big Data (Code: KEN4254)

Examiner: Dr. M. Mihalák

Desired prior knowledge: Discrete mathematics, algorithm design and analysis, elementary discrete probability

Prerequisites: None

Description: The emergence of very large datasets poses new challenges for the algorithm designer. For example, the data may not fit into the main memory anymore, and caching from a hard-drive becomes a new bottleneck that needs to be addressed. Similarly, algorithms with larger than linear running time take simply too long on very large datasets. Moreover, simple sensory devices can observe large amount of data over time, but cannot store all the observed information due to insufficient storage, and an immediate decision of what to store and compute needs to be made. Classical algorithmic techniques do not address these challenges, and a new algorithmic toolkit needs to be developed. In this course, we will look at a number of algorithmic responses to these problems, such as: algorithms with (sub-)linear running times, algorithms where the data arrive as a stream, computational models where memory is organized hierarchically (with larger storage units, such as hard-drives, being slower to access than smaller, faster storage such as CPU cache memory). New programming paradigms and models such as MapReduce/Hadoop will be discussed. We will also look at a number of topics from classical algorithm design that have undiminished relevance in the era of big data such as approximation algorithms and multivariate algorithmic analysis.

Knowledge and understanding: Students will know, exemplified on selected topics, what can be provably achieved when designing and analysing algorithms for very large datasets, and will know some of the most successful state-of-the-art algorithmic techniques for dealing with algorithmic challenges posed by large data sets.

Applying knowledge and understanding: Students will be able to adjust and apply the gained knowledge about algorithmic techniques to various algorithmic challenges of handling large data-sets.

Making judgements: Students will be able to categorize large-scale problems according to their computational feasibility, and select the appropriate algorithmic response.

Communication: Students will be able to reason about computational problems and algorithms addressing the problems in a clear, exact, and unambiguous way.

Learning skills: Additionally to the guiding material provided by the lecture, the students will autonomously search, read, and study the details from various sources.

Study material: Will be provided throughout the lecture.

Recommended literature: None.

Exam: Written exam (80%) at the end of the course and graded exercises (20%) throughout the course.

ECTS: 6
Dynamic Game Theory (Code KEN4251)

**Examiner:** Prof. dr. F. Thuijsman.

**Desired Prior Knowledge:** Students are expected to be familiar with basic concepts from linear algebra, calculus, Markov chains and differential equations.

**Prerequisites:** None

**Description:** The course will focus on non-cooperative games and on dynamic games in the following order: matrix and bimatrix games, repeated games, Stackelberg games, differential games, specific models of stochastic games, evolutionary games. These are games in which the players are acting as strategic decision makers, who cannot make binding agreements to achieve their goals. Instead, threats may be applied to establish stable outcomes. Besides, relations with population dynamics and with “learning” will be examined. Several examples will be taken from biological settings.

**Knowledge and understanding:** Students are able to recognize and classify the main types of dynamic games, i.e. repeated games, stochastic games, Stackelberg games, differential games, and evolutionary games and formulate the main solution concepts value, optimal strategies, Nash- and Stackelberg equilibrium.

**Applying knowledge and understanding:** Students are able calculate solutions of the different types of dynamic games.

**Making Judgements:** Students are able to explain advantages and disadvantages of different solution concepts. They are able to judge correctness of solutions presented.

**Communication:** Students are able to explain and defend correctness of their solutions.

**Learning Skills:** By the end of the course, students will be able to autonomously and critically reflect upon the pros and cons of different types of games for modelling competition and cooperation. This includes considerations on the computational aspects with respect to different solution concepts.

**Study material:** Handouts will be provided.

**Recommended Literature:** none.

**Exam:** There will be a closed book written exam at the end of the course.

**ECTS:** 6
Computational Statistics (KEN4258)

Examiner: Dr. C. Seiler

Desired prior knowledge: Probability and Statistics (Code: KEN2130)

Prerequisites: None

Description: Complex and high dimensional data are abundant in academia and industry. At the same time, computers are cheap and powerful. These developments enable us to fit increasingly complex statistical models using computer intensive methods. We will model and analyze both independent and dependent data from real world problems. The course is hands-on; we will use Stan (platform for statistical modelling and high-performance statistical computation) and R (statistical programming language). Key topics: Statistical modelling, uncertainty quantification, Markov chain Monte Carlo, bootstrap resampling, permutation tests, and causal inference.

Knowledge and understanding: Knowing a wide range of modern statistical models and computational tools to draw inferences will provide the foundations for analyzing complex data in academia and industry.

Applying knowledge and understanding: Students will be able to:
1. Tidy raw data obtained from large databases
2. Build complex statistical models
3. Choose computational tools to perform inference
4. Create reproducible analysis workflows to communicate results

Making judgements: In this course, we will discuss one of the most important aspects of analyzing data: being skeptical of results and avoiding wishful thinking. This will be accomplished by careful model checking and interpretation of the results.

Communication: Students will present their results using literate programming and reproducible workflows.

Learning skills: Students will be able to understand, apply, and extend papers from computational statistics journals.

Study material:

Recommended literature:

Selected chapters:
- Efron and Tibshirani (1993), An Introduction to the Bootstrap
- Hoff (2009), A First Course in Bayesian Statistical Methods
- Grolemund and Wickham (2017), R for Data Science
- Hernán and Robins (2019, forthcoming), Causal Inference

Exam: Written exam and assignments (focus on programming exercises, only few selected mathematical exercises).

ECTS: 6
Advanced Natural Language Processing (KEN4259)

**Examiner:** Dr. J. Niehues

**Desired prior knowledge:** Advanced Concepts in Machine Learning

**Prerequisites:** none

**Description:** How do I say, “Where is the next Italian restaurant” in Dutch? Can I get a summary of today’s lecture? When were artificial neural networks developed? Computers able to answer these questions are a long-time dream of humankind and currently, we see first programs to solve these problems. This course will provide the skills and knowledge to develop state-of-the-art (SOTA) solutions for these natural language processing (NLP) tasks.

After a short introduction to traditional statistical approaches to NLP, the course will focus on deep learning techniques to solve these problems. In the first part of the course, we will investigate methods to model sequence labeling tasks like Named Entity recognition or Part-of-speech techniques. The second part of the lecture will focus on sequence-to-sequence models, a very powerful model to solve many NLP tasks like machine translation, summarization and question answering.

In this course, major challenges when building the systems will be address: representing words in neural networks, neural network architectures to model language, methods to train complex models and algorithms to find the most probable output.

**Knowledge and understanding:** Student will be taught state-of-the-art deep learning techniques for natural language processing, especially sequence labeling and sequence-to-sequence models. They will learn techniques to address the major challenges when building a natural language processing tool.

**Applying knowledge and understanding:** The achievements in deep learning have significantly improved the quality of state-of-the-art methods for natural language processing. With the knowledge acquired in the course, students will be able to build SOTA solutions.

**Making Judgements:** Students will be able to analyze the specific challenges of a task in NLP. Based on the gather knowledge on different ways to model tasks they are able to select and implement a fitting model to solve the task.

**Communication:** Through small research projects, students will be enabled to communicate their findings and explain the rationale behind their choices in deep learning techniques for natural language processing.

**Learning Skills:** After successful completion of the course, students will be able to develop natural language processing tools and perform research on new ideas in the field.

**Study material:** Mostly based on the lecture notes and the provided material including recent papers published in this field.

**Recommended literature:** Papers published in top international conferences and journals in machine learning field.

**Exam:** Group-assignment (30%), final exam (70%)

**ECTS:** 6
Building and Mining Knowledge Graphs (Code: KEN4256)

Examiner: Prof. dr. Michel Dumontier and dr. Kody Moodley

Tutor(s): None.

Desired Prior Knowledge: Introduction to Computer Science

Prerequisites: None.

Description: Knowledge graphs are large-scale, machine-processable representations of entities, their attributes, and their relationships. Knowledge graphs enable both people and machines to explore, understand, and reuse information in a wide variety of applications such as answering questions, finding relevant content, understanding social structures, and making scientific discoveries. However, the sheer size and complexity of these graphs present a formidable challenge particularly when mining across different topic areas.

In this course, we will examine approaches to construct and use knowledge graphs across a diverse set of applications using cutting-edge technologies such as machine learning and deep learning, graph databases, ontologies and automated reasoning, and other relevant techniques in the area of data mining and knowledge representation.

Knowledge and understanding: Students will be able to describe:
- The nature and attributes of a Knowledge Graph
- Examples of Knowledge Graphs;
- Representations for Knowledge Graphs
- Applications of Knowledge Graphs
- Advantages and disadvantages of Knowledge Graphs as compared to other formalisms
- Approaches and challenges in constructing and maintaining Knowledge Graphs
- Approaches and challenges in finding, using and mining Knowledge graphs
- What FAIR is and how it relates to Knowledge Graphs
- Ethical, legal & social issues around Knowledge Graphs

Applying knowledge and understanding: Students will be able to identify requirements and steps to convert knowledge in traditional data formats to Knowledge Graph formats. Students will also be able to implement such strategies. Students will be able to query Knowledge Graphs (for instance using SPARQL query language) to answer basic to intermediately advanced questions. Students will be able to implement basic reasoning strategies on Knowledge Graphs to answer intermediately advanced questions, which cannot be answered by SPARQL queries alone. Students will be able to implement popular methods to integrate different data sources by transferring them into a Knowledge Graph. Students will be able to enrich existing Knowledge Graphs with missing information using basic predictive algorithms. Students will be able to perform basic data quality assessment on Knowledge Graphs. Students will be able to assess the degree of compliance that Knowledge Graphs have with FAIR principles.

Making judgements: Students will be able to select which tools are most suitable for constructing, querying, visualising & reasoning with Knowledge Graphs. Students will be able to differentiate between different types of Knowledge Graphs, according to their representation, coverage and content. Students will be able to select which Knowledge Graph is appropriate for answering a particular question. Students will be able to diagnose incompleteness in a Knowledge Graph with respect to answering a particular question. Students will be able to evaluate the data quality and FAIRness of a Knowledge Graph.

Communication: Students will be able to explain the advantages of representing information on the web in Knowledge Graphs. Students will be able to communicate the steps required to convert information to a Knowledge Graph format. Students will be able to communicate to non-experts the main content and representational components of a Knowledge Graph. Students will be able to outline to non-experts the steps required to answer a question by querying a Knowledge Graph.

Learning Skills: Students will be able to reflect critically on the challenges and open problems remaining in Knowledge Graphs research. Students will be able to formulate and propose strategies to answer complex questions using Knowledge Graphs. Students will be able to assess the feasibility of different combinations of methods for answering questions using Knowledge Graphs.

Study material: Material will be provided during the course in the form of handouts.


Exam: Individual project for application of knowledge and two assignments to demonstrate understanding of core concepts.

ECTS: 6
Data Fusion (KEN4223)

Examiners: prof. Anna Wilbik

Desired prior knowledge: statistics and basic machine learning

Prerequisites: none

Course description: ICT development, e.g., remote sensing, IoT, lead to an enormous growth of available data for analysis. To integrate this heterogeneous or multimodal data, data fusion approaches are used. Data fusion can be understood as a framework for the joint analysis of data from multiple sources (modalities) that allows achieving information/knowledge not recoverable by the individual ones.

During this course, several approaches to data fusion will be discussed, such as:

(1) Low level data fusion, where data fusion methods are directly applied to raw data sets for exploratory or predictive purposes. A main advantage is the possibility to interpret the results directly in terms of the original variables. An example of a low level data fusion is measuring the same signal or phenomena with different sensors, in order to discover the original one. Traditionally, PCA based methods are used for this type of data fusion.

(2) Mid level data fusion, where data fusion operates on features extracted from each data set. The obtained features are then fused in a “new” data set, which is modeled to produce the desired outcome. A main advantage is that the variance can be removed in the features extraction step, and thus the final models may show better performance. An example of a mid level data fusion is extracting numerical features from an image, and building a decision model based on those features.

(3) High level data fusion, also known as decision fusion, where decisions (models outcome) from processing of each data set are fused. It is used when the main objective is to improve the performance of the final model and reach an automatic decision. Several methods can be used for high-level DF, such as weighted decision methods, Bayesian inference, Dempster Shafer’s theory of evidence, and fuzzy set theory. There is a link between high-level data fusion and ensemble methods.

(4) Federated learning. Federated learning enables multiple parties jointly train a machine learning model without exchanging the local data. In case of federated learning, we can talk about model fusion.

Knowledge and understanding: The student can explain fusion on the different levels: low level, mid level and high level as well as federated learning. They can identify which approach is appropriate for a problem in hand.

Applying knowledge and understanding: Students are able to describe the advantages and disadvantages of different methods. Students have obtained the knowledge to develop, program, analyse, and apply fusion methods to a wide variety of problems in the context of data-driven decision making. Making judgements: Students will be able to judge the quality of models, results and approaches (e.g., scientific publications). Communication: Students will be able to present the results the fusion models to specialists or non-specialists.

Learning skills: Students will be able to familiarize themselves with fusion techniques beyond the scope of the course in order to solve a problem.

Study material: Course notes and other information made available.

Exam: Written exam (70%) + assignment (30%)

ECTS: 6
Period 1.5

Computer Vision (KEN4255)

Examiner: Dr Stylianos (Stelios) Asteriadis and dr. M. Popa

Desired prior knowledge: Basic knowledge of Matlab, linear algebra and machine learning. This course offers the basics on image processing although prior knowledge is also a plus.

Description: Can we make machines look, understand and interpret the world around them? Can we make cars that can autonomously navigate in the world, robots that can recognize and grasp objects and, ultimately, recognize humans and communicate with them? How do search engines index and retrieve billions of images? This course will provide the knowledge and skills that are fundamental to core vision tasks of one of the fastest growing fields in academia and industry: visual computing. Topics include introduction to fundamental problems of computer vision, mathematical models and computational methodologies for their solution, implementation of real-life applications and experimentation with various techniques in the field of scene analysis and understanding. In particular, after a recap of basic image analysis tools (enhancement, restoration, color spaces, edge detection), students will learn about feature detectors and trackers, fitting, image geometric transformation and mosaicing techniques, texture analysis and classification using unsupervised techniques, object classification and face recognition, camera models, epipolar geometry and 3D reconstruction from 2D views.

Knowledge and understanding: Students will be able to apply the most suitable techniques for image pre-processing (e.g. enhancement, restoration), feature extraction, texture analysis, perspective geometry, camera models and topics on object recognition. Students will be able to identify the most suitable techniques in a series of visual computing problems

Applying knowledge and understanding: The students will be able to choose and/or construct solutions in a variety of professional/vocational contexts requiring image processing and computer vision (robotics, manufacturing, AI, web applications, surveillance). They will be able to build and assess methodologies for handling real-world complex problems in computer vision, making use of pre-existing data for training their models.

Making judgements: The students will be able to choose and combine the right methods to tackle real-world computer vision problems, captured in real-life settings and having no obvious solutions. They will be able to propose and build techniques combining computer vision methods along with machine learning instruments for scene understanding and object recognition.

Communication: Through small research projects, students will be able to communicate their findings and explain the rationale behind their choices in computer vision techniques for image/video analysis.

Learning skills: After successful completion of the course, students will be able to analyse images and videos and retrieve or process content in order to derive useful information, applicable in a variety of domains (e.g. satellite imagery, surveillance, robotics, medical imaging)

Study material:

- Lecture slides and provided notes

Exam: Projects (33% individual assignments – 17% group-based assignments) and final exam (50%).

ECTS: 6
Information Retrieval and Text Mining (Code: KEN4153)

Examiner: Prof dr. ir. J.C. Scholtes
Tutor(s): None.
Desired Prior Knowledge: None.
Prerequisites: None.

Description: Text mining refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining encompasses several computer science disciplines with a strong orientation towards artificial intelligence in general, including but not limited to information retrieval (building a search engine), statistical pattern recognition, natural language processing, information extraction and different methods of machine learning, clustering and ultimately data visualization. An important difference with standard information retrieval (search) techniques is that they require a user to know what he or she is looking for, while text mining attempts to discover information in a pattern that is not known beforehand. This is very relevant, for example, in criminal investigations, legal discovery, (business) intelligence, sentiment- & emotion mining or clinical research.

Knowledge and understanding: The student will be able to select, understand and apply different phases and methods used to create successful Information Retrieval and Text Mining applications. In addition, the student learns to evaluate the quality of such methods according to best-practice standards as used in the field.

Applying knowledge and understanding: Students will be able to recognize applications of text mining and information retrieval in different domains such as legal services, medical research, regulatory oversight, compliance, humanities, and customer services. After the course, the student can formulate an opinion or course of action when dealing with text-based KE-problems based on incomplete, limited and in part unreliable information. After the course, students can apply their knowledge and understanding in a manner that shows a scientific approach to their work or vocation. They are able to handle complex and ill-defined text-based problems for which it is not a priori known if there is an appropriate solution, they know how to acquire the necessary information to solve the sub-problems involved, and they know how to proceed with problems for which there is no standard or reliable route to the solution.

Making judgements: Upon completion of the course, students are able to recommend the most appropriate methods from the fields of text mining and information retrieval when confronted with KE-problems involving textual and other forms of unstructured data.

Communication: Students are able to communicate the (dis)advantages of several methods from the field of text mining and information retrieval to both an audience of non-experts.

Learning skills: After the course, the student has developed those learning skills that are necessary for a successful further career in text mining or information retrieval at the highest professional level. The student will be able to continue to develop their text-mining and information retrieval skills. The student is able to detect missing knowledge and abilities and to deal with them appropriately by finding and consulting resources that can help them to fill the gaps and new developments.

Study material: A syllabus and copies of the course slides will be used.


Exam: The result of the practical exercises contributes 30% to the final examination of the course. The other 70% is determined by the theoretical exam. The theoretical exam is open book. For the practical exercise, students can select a research topic and a text corpus from the provided list (or another relevant open source collection) and implement a number of relevant text-mining operations by using open source text-mining tools. A number of relevant pre-processing operations, text mining operations, and visualizations have to be implemented. Proposals of work will have to be within one week after the start of the course, after which they will be reviewed. After approval, the students can start the implementation of their proposals. At the end of the course, each student or group shall write a report on the research and the results and the results shall be presented to the rest of the class.

ECTS: 6
Deep Learning (Code: KEN4257)

Examiners: Dr. Siamak Mehrkanooon

Desired Prior Knowledge: Machine Learning

Prerequisite: Advanced Concepts of Machine Learning

Course description: Conventional machine learning techniques were limited in processing data in their raw forms and many domain experts were required in transforming raw data into meaningful features or representations. Deep Learning techniques have revolutionized many application domains ranging from auditory to vision signal processing. In this course, we will study various concepts in deep architectures using both artificial neural networks as well as kernel-based models. Several deep learning models such as convolutional neural networks, auto-encoders, generative adversarial networks and their variants among other state-of-the-art models will be covered in depth. We will further study different types of deep architectures used for domain adaptation problems where one is encountered with heterogeneous datasets as well as multi-modal datasets. The regularization and optimization methods used in deep learning framework will be discussed. Introduction to open-source deep learning platforms will be given. This course will be equipped with a practical component, and students are expected to write their own deep learning code and test its performance on various problems. In addition they are strongly encouraged to participate in mini-projects (in a group or individual) targeting a conference paper.

Knowledge and understanding: By the end of this course, students will be able to explain aspects related to learning the model parameters, overfitting, model architecture, relation between neurons, layers among others. In addition, they will be able to implement the methodologies using deep learning libraries. The learned deep machine learning models combined with mathematics and optimization behind the models will enable students to gain more insight on capabilities of these models.

Applying knowledge and understanding: Students will be able to apply the above-mentioned deep learning skills, which they have acquired in their domain of interest. Explore and select new research directions and discover new things by driving new models.

Making judgements: Students will be able to implement and apply deep learning models for analyzing datasets in different tasks. They will be able to determine the best model for the given task.

Communication: Students will be able to discuss deep learning models and their results with scientists, engineers and both expert and non-expert.

Learning skills: By the end of the course, students will be able to autonomously follow up the recent trends in deep learning which are beyond the scope of the provided course materials.

Study material: Course notes and other information made available.

Recommended Literature:

• Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning, MIT Press.
• Research Papers Published in high ranked journals and conferences.

Exam: Grades of the assignments (20%) + Grade for paper review, presentation (20%) + Grade for the exam (60%). Bonus: If students can come up with novel approaches to solve the given assignments and after consulting the results with me they can write a manuscript describing their findings and submit it to a conference for review get additional 10% bonus on the assignments which could help them for the final grades. This bonus does not depend on the acceptance of the manuscript by the conference.

ECTS: 6
Planning and Scheduling (Code KEN4253)

**Examiner:** Dr. M. Mihalák

**Desired prior knowledge:** Data Structures & Algorithms. Discrete Mathematics. Graph Theory

**Prerequisites:** None

**Description:** In many real-world processes, particularly in industrial processes and logistics, decisions need to be taken about the time of the completion of (sub)tasks, and the decision about what production machines complete the tasks. There are often constraints on the order in which tasks, or ‘jobs’ can be performed, and there are usually capacity constraints of the machines. This leads to natural, industrially critical optimization problems. For example, a company might choose to buy many machines to process jobs, but then there is a risk that the machines will be underused, which is economically inefficient. On the other hand, too few machines, or an inappropriate ordering of tasks, may lead to machines spending a significant amount of time standing idle, waiting for the output of other machines, which are overcrowded with tasks. In this course, we look at various mathematical models and techniques for optimizing planning and scheduling problems, subject to different optimality criteria. We will discuss, among others, single-machine models, parallel-machine models, job-shop models, and algorithms for planning and scheduling (exact, approximate, heuristic) and we also touch upon the computational complexity (distinguishing between ‘easy’ and ‘difficult’ problems) of the underlying problems. Last but not least, we will also introduce integer linear programming as a uniform and generic tool to model and solve planning and scheduling problems.

**Knowledge and understanding:** Students will possess the mathematical and algorithmic tools to model and solve planning/scheduling problems. Students will be able to recognize real-world problems in the unified theory and established language of planning and scheduling.

**Applying knowledge and understanding:** Students will be able to apply the new techniques to various problems arising in real-world applications. Students will be able to deploy the standard algorithmic techniques, and be able to design new algorithmic solutions, and to argue about their performance properties.

**Making judgements:** Students will understand under which circumstances different planning/scheduling problems are computationally tractable, and will judge algorithmic technique can be used to exactly or approximately solve these problems.

**Communication:** Students will be able to analytically argue about correctness of the used algorithmic approaches. Students will be able to explain modelling approaches to planning and scheduling problems in the language of the theory of planning and scheduling.

**Learning skills:** Students will enhance their study skills such as time management, effective reading, critical thinking and reading, exact and unambiguous writing and formulating of ideas and statements, and reflection on marked work. Along the way, students will improve general learning skills such as self-motivation, careful listening and giving instructions, and openness to new knowledge. Students will also be exposed to autonomous self-study.

**Study material:** Appropriate study material will be provided throughout the course.

**Recommended literature:** None

**Exam:** Written exam (80%) at the end of the course, and graded exercises (20%) throughout the course.

**ECTS:** 6
Explainable Artificial Intelligence (Code: 4246)

Examiners: Prof. Dr. N. Tintarev and Dr. Tjitze Rienstra
Prerequisites: Data Mining or ACML
Desired Prior Knowledge: Data Analysis

A key component of an artificially intelligent system is the ability to explain to a human agent the decisions, recommendations, predictions, or actions made by it and the process through which they are made. Such explainable artificial intelligence (XAI) can be required in a wide range of applications. For example, a regulator of waterways may use a decision support system to decide which boats to check for legal infringements, a concerned citizen might use a system to find reliable information about a new disease, or an employer might use an artificial advice-giver to choose between potential candidates fairly. For explanations from intelligent systems to be useful, they need to be able to justify the advice they give in a human-understandable way. This creates a necessity for techniques for automatic generation of satisfactory explanations that are intelligible for users interacting with the system. This interpretation goes beyond a literal explanation. Further, understanding is rarely an end-goal in itself. Pragmatically, it is more useful to operationalize the effectiveness of explanations in terms of a specific notion of usefulness or explanatory goals such as improved decision support or user trust. One aspect of intelligibility of an explainable system (often cited for domains such as health) is the ability for users to accurately identify, or correct, an error made by the system. In that case it may be preferable to generate explanations that induce appropriate levels of reliance (in contrast to over- or under-reliance), supporting the user in discarding advice when the system is incorrect, but also accepting correct advice.

The following subjects will be discussed:
(1) Intrinsically interpretable models, e.g., decision trees, decision rules, linear regression.
(2) Identification of violations of assumptions; such as distribution of features, feature interaction, non-linear relationships between features; and what to do about them.
(3) Model agnostic explanations, e.g., LIME, scoped Rules (Anchors), SHAP (and Shapley values)
(4) Ethics for explanations, e.g., fairness and bias in data, models, and outputs.
(5) (Adaptive) User Interfaces for explainable AI
(6) Evaluation of explanation understandability

Knowledge and understanding: Students can explain the difference between different explanation approaches (e.g., global versus local models) and can identify which are suitable to use based on underlying assumptions and relative advantages and limitations.
Applying knowledge and understanding: Students can critically choose and apply XAI methods. Students can formulate evaluation protocols to validate the understandability of explanations, demonstrating awareness of the ethical, normative, and social consequences of their applications.
Making judgements: Students will be able to critically evaluate the quality (rigor of methodology), and ethical consequences, of approaches (systems or scientific publications) based on the XAI techniques taught.
Communication: Students will be able to communicate their ideas effectively in written form. They will be able to actively contribute to group-wise communication, and in both oral and written form present their models and outputs to specialists.
Learning skills: Students will be able to familiarize themselves, and critically assess XAI techniques beyond the scope of the course in order to solve a problem.
Study material: Course notes, required reading of scientific articles.

Recommended Literature:
• Rothman, Denis. Hands-On Explainable AI (XAI) with Python: Interpret, visualize, explain, and integrate reliable AI for fair, secure, and trustworthy AI apps, Packt, 2020
Exam: Active reading (20%) + two written assignments (2*25% = 50%) + group project (30%)
ECTS: 6
Symbolic Computation and Control (Code: KEN4252)

Examiner: Prof dr. ir. R.L.M. Peeters.

Desired Prior Knowledge: Linear Algebra, Calculus, Mathematical Modelling.

Course description: This course consists of two interrelated parts. In the first part, we focus on basic techniques for the digital control of linear dynamical systems using feedback. We start by addressing system stability and we discuss the technique of pole placement by state feedback to solve the regulation problem. Then we introduce state observers to solve the regulation problem by output feedback. Next, we extend our scope to tracking problems. This involves the design of additional dynamics to characterize the relevant class of reference signals, which are then integrated with the earlier set-up for output feedback. Finally, we discuss the classical topic of optimal control, which can be employed to avoid using prototype systems for pole placement, and which allows the user to design a feedback law by trading off the cost involved in generating large inputs against the achieved tracking accuracy. In the second part, we address computational issues, related to the field of systems and control. Classically, computers have been designed primarily to perform approximate numerical arithmetic. Modern software packages for mathematical computation, such as Maple and Mathematica, allow one to perform exact and symbolic computation too. We shall explore this new area. It is demonstrated how speed, efficiency and memory usage considerations often lead to surprising and fundamentally different algorithmic solutions in a symbolic or exact context. Applications and examples involve stability of linear systems, model approximation, and linear matrix equations with free parameters. Practical classes serve to demonstrate the techniques and to make the student familiar with exact and symbolic computation.

Knowledge and understanding: Students familiarize themselves with state and output feedback to achieve control of dynamical systems. Concretely, they learn to (mathematically) build a basic stabilizing feedback controller for a linear input-output dynamical system, using a combination of different design techniques. Students learn methods for exact numerical and symbolic computation, as used in algebraic computation with unspecified parameters. They also learn in which ways these are different from the more commonly used approximate numerical (floating-point) methods: in terms of accuracy, speed (complexity), and memory usage.

Applying knowledge and understanding: Students will be able to construct and implement, for a given linear dynamical input-output system: (a) stabilizing state feedback, (b) full state observer, and (c) additional dynamics to perform tracking of a specified output trajectory. They will also be able to assess the quality of a controller, regarding an optimal control LQ criterion, and in view of the desired settling time and the trajectory approximation. Students will be able to determine the stability of a given linear dynamical system in an exact and/or symbolic algebraic way. They will also be able to efficiently solve linear systems of (matrix) equations involving symbolic parameters, avoiding pitfalls, which arise from techniques from approximate numerical computation.

Making Judgements: Students will be able to judge the quality of a feedback design for stabilization (regulation) or tracking. Students will be able to indicate which exact and symbolic computation methods will and will not be useful for a given parameterized problem, regarding speed and memory usage.

Communication: Students will be able to motivate the design of a feedback controller, the construction of a trajectory approximation, the design of a full state observer, and the implementation choices of the weights in LQ-design. They will be able to explain the concept of feedback in the area of control. Students can adequately discuss speed and efficiency properties of an algorithm (approximate numerical, exact numerical, symbolic) to specialists and non-specialists.

Learning Skills: Students will be able to read and interpret basic scientific literature on control theory and on numerical and symbolic computation. They can use Matlab and the Control Toolbox and work out ideas computationally. Students can use some of the exact and symbolic functionality of Mathematica and work out ideas computationally.

Study material: Syllabus, provided on eleUM. Handouts.


Exam: Written exam by computer in two parts, each having a weight of 50% on the final grade: one midterm take-home exam with Matlab on part 1 (control), one final classroom exam with Mathematica on part 2 (symbolic computation). The resit exam is on both parts of the course in a classroom setting.

ECTS: 6
Algorithms for Data Visualization (KEN4213)

Examiner: Dr. S. Chaplick

Desired prior knowledge: Data Structures & Algorithms. Discrete Mathematics. Graph Theory

Prerequisites: None

Description: In our modern world, we are surrounded by data sets in all shapes and sizes. An essential aspect of working with data sets (whether relational, quantitative, etc.) is how they should be presented/visualized. Even for a single data set, different visualizations can be better for different tasks. Moreover, the scale of the data sets often restricts the options available when designing how it should be presented (including the choice of algorithm and the appropriateness of preprocessing/cleaning the data to a “visualizable” scale). This course will provide an overview of the basic theoretical and practical aspects of information visualization with a focus on algorithmic approaches. It includes how to visualize relational data (e.g., graphs/networks) and standard approaches for quantitative data sets such as, projecting high dimensional data to lower dimensions for visualisation (e.g., multi-dimensional scaling and t-distributed stochastic neighbour embedding, etc.). We will also cover some aspects of augmenting visualizations with meta-data such as, labeling nodes/points, weighting relations, and information regarding grouping/clustering.

Knowledge and understanding: Students will be able to identify and recount the common approaches used in information and network visualization. For example, students will be able to identify visualization techniques used in systems/media. Moreover, students have a sufficient understanding of such these approaches to implement them.

Applying knowledge and understanding: Students will be able to adapt existing and design new visualization approaches to various problems arising in real-world applications. Students will be able to design efficient implementations of such approaches via standard algorithmic techniques, and to argue about their performance.

Making judgements: Students will be able to evaluate under which circumstances different visualization techniques are viable, and to select an appropriate technique to be applied depending on the setting/task.

Communication: Students will be able to analytically argue about correctness and performance of the used algorithmic approaches. Students will be able to explain and compare different visualization approaches.

Learning skills: By the end of the course, students will be able to autonomously evaluate a new visualization technique as it relates to standard approaches. For example, students will also be able to read and discuss visualization literature beyond the scope of this course (as showcased via a short presentation during the course).

Study material: Appropriate study material will be provided throughout the course.

Recommended literature:

Exam: Written exam (80%) at the end of the course, graded exercises (10%) throughout the course, and a short presentation (10%).

ECTS: 6
**Period 1.6**

Research Project AI 2 & DSDM 2 (Code KEN4131 & KEN4231)

**Examiner:** T.b.a.

**Coordinator:** Dr. ir. Kurt Driessens

**Desired Prior Knowledge:** None.

**Prerequisites:** None.

**Description:** The research project takes place during the three periods of the semester. Project topics are presented at the start of the semester and assigned to students based on their preferences and availability. The emphasis in the first phase is on initial study of the context set out for the project and the development of a project plan. In the second period, the goal is to start modelling, prototyping and developing. In period 3, the implementation, model and/or experiments set out in the project plan has to be finished and reported on. At the end of period 1 and 2, a progress presentation takes place. The project results in a project presentation, a project report and possibly a public website and/or product.

**Knowledge and understanding:** Students get to know and possibly contribute to state of the art methods within the fields of Artificial Intelligence and/or Data Science for Decision Making to answer an open question.

**Applying knowledge and understanding:** Student write their own research plan in coordination with a staff member (plus possibly outsiders) who act as clients with an open question. Students with different backgrounds and from both masters work together in teams to build and evaluate an answer to an open question. Students find, judge the suitability, apply, and evaluate state of the art techniques to answer questions and construct applications in the field of Artificial Intelligence and Data Science. Students apply the accumulated knowledge from other educational activities in application specific areas.

**Making judgements:** Students judge feasibility of tasks, attainability of goals, and the amount of work involved. Students think about the possible consequences of their work. Students evaluate state of the art and the applicability and scope of research results.

**Communication:** Students will learn to:

1. orally communicate and cooperate with peers
2. orally report on progress and intermediate results to superiors
3. orally negotiate and communicate with clients
4. communicate their ideas in written form, both for an academic and a general audience
5. give effective presentations

**Learning Skills:** Students increase their own level of knowledge in a specialised sub-discipline of the field of Artificial Intelligence and/or Data Science. Students perform research into recent state of the art techniques. Students learn that the field of Artificial Intelligence and Data Science are constantly evolving beyond what is taught in class.

**Study material:** Slides provided at the end of joint information sessions. Literature provided by the project supervisors.


**Exam:** Phase 1: project plan + presentation (15%); Phase 2: layman’s website + presentation (15%); Phase 3: Project report + presentation (70%)

**ECTS:** 6
3.2 Year 2 of the Master’s Programmes AI and DSDM

Period 1, 2 and 3 of year two of the master’s program consist of electives to be chosen by the student. This optional program can be assembled freely at your own choice, but within academic significance and relevance to your master’s track. The Board of Examiners has to evaluate and approve the chosen combination of electives. The electives consist of the following options to choose from: courses to be followed at DKE, at other UM Master programmes, at another university, a research project, an internship, a study abroad at a foreign university, or a project. Note that you have obtained at least 40 ECTS of course year 1 in order to enter the second year of the programme.

Electives at Maastricht University outside DKE

It is possible to take electives at other relevant master’s programmes at Maastricht University for at most 13 ECTS in the second year of the programme. The following courses below will be automatically approved by the Board of Examiners of DKE. You should apply through the Special Course Approval procedure via the My UM Portal. Note that they may have limited capacity.

School of Business and Economics
Social Choice Theory (ECB4005) 6.5 ECTS
Supply Chain Operations (EBC4016) 6.5 ECTS
Negotiation and Allocations (EBC4193) 6.5 ECTS
Intellectual Property Rights in a Digital Economy (EBC4026) 6.5 ECTS
High-Dimensional Econometric Methods for Big Data (EBC4218) 6.5 ECTS

Faculty of Psychology and Neuroscience
Auditory and Higher Order Language Processing (PSY4051) 4 ECTS
Perception and Attention (PSY4052) 4 ECTS
Sensorimotor Processing (PSY4055) 4 ECTS

Exam: Depends on content of the elective program.
ECTS: 30

Besides complying that you have passed 40 ECTS, for taking these electives at FPN you should have passed “Advanced Concepts in Machine Learning” and “Autonomous Robotic Systems” at DKE.

Internships:
Another option for the elective semester in the Master Programme is to conduct a Business or Research internship. The students can choose the company or research organisation themselves. Together with a DKE supervisor and a representative of the host organisation, the student fills out an internship proposal (which can be found here on Student Portal) and this requires approval of the Board of Examiners prior to its start. For more information about doing an internship, look at the FAQ sheet here.
3.3 Master’s thesis AI & DSDM (Code: KEN4160 & Code: KEN4260)

The Master’s Artificial Intelligence and Data Science for Decision Making will be completed by writing a master’s thesis. The thesis is produced individually and is the result of a master’s research project that runs during the second semester of year 2 of the master’s programme. In the first phase, the emphasis is on self-study, subject determination, planning and some preliminary research. Then the actual research is started. The final phase is used to finalize the master’s thesis. The master’s project is completed by a presentation of the results. The master’s project will be supervised by one of the senior researchers.

Exam: Master’s thesis and presentation.
ECTS: 30

* Note that when you enrol in February, you follow your electives in period 2.4, 2.5 and 2.6 and work on your master’s thesis in period 2.1, 2.2, 2.3.

Master’s thesis Artificial Intelligence and Data Science for Decision Making

At the end of the master’s study, each student has to write a thesis. This thesis has to be written and presented individually, and accompanied by relevant attachments and software. In order to start working on the thesis, a student needs to have obtained at least 60 ECTS (among which are 40 credits of the first year).

General procedure
The process of writing a master’s thesis consists of 6 phases. It is scheduled in the last semester of the master’s study. The time frame given below is an indication for these phases.

Phase 1: Topic selection
At the end of the previous semester, the students are informed of the main directions of research in the three research groups at the DKE department. Based on this information, students acquire more information about specific possibilities in the areas by means of individual discussions with relevant DKE researchers available. These discussions take place upon the initiative of the student.

Phase 2: Thesis Research Plan
The student creates a thesis research plan, which is to be signed by the student and the thesis supervisor, and then handed over to the master’s thesis coordinator. The plan is sent to the Board of Examiners for approval. In the fourth week of period 1, the students present their research topics during a 10 minutes presentation in front of the DKE staff members and fellow students.

Phase 3: Research
During the second period of the semester, the student carries out his/her own research. This research process will be guided by the thesis supervisor through a series of frequent appointments, preferably on a weekly basis. In the fourth week of period 2, the student is invited to present the first research achievements, with audience the DKE staff members and fellow students.

Phase 4: Writing
At the end of the second period, the research is finished and the first version of the thesis is submitted to the thesis supervisor. The thesis supervisor, as well as a second assessor, will evaluate the thesis and provide a first reaction within one week. The second and final submission will take place during the second week of the third period.
Phase 5: Preparation for presentation
In the third week of the third period, the student prepares a final presentation of the thesis research. This individual presentation will have a maximum length of 30 minutes, followed by 15 minutes of discussion.

Phase 6: Presentation
The presentation is open for all students and teaching staff of the bachelor’s Data Science and Artificial Intelligence and master’s programmes, Artificial Intelligence and Data Science for Decision Making and anyone else who was involved in the thesis research. The final decision on the grade for the master’s thesis will be made shortly after the presentation.

Requirements and assessment
For the master’s thesis research, every student has to conduct a short scientific research. This can be empirical or theoretical research. The topic is open, as long as it fits into the field of the master’s program. DKE staff will briefly introduce their main areas of research, but students are encouraged to propose a research topic themselves. The topic and the research question have first to be approved by the thesis supervisor. To achieve this, the student will create a master’s thesis research plan using the form provided by the Board of Examiners that contains amongst others:

- Date;
- Name and student number;
- Name of the thesis supervisor(s) and examiners;
- Title of the master’s thesis, start and end date of the thesis research;
- Short description of the problem statement and research questions;
- A list of the main research activities, with a time schedule.

This plan will be signed by the student and the thesis supervisor and prospective examiners and then handed in to the Board of Examiners for the formal approval.

It is possible to execute the master’s thesis research as an external training period. This should be well defined in the master’s research plan. In this case, the plan should also include the name of the company, the name of the external supervisor, the size of the project and any agreements about payment and confidentiality. The plan should also be signed by the external supervisor.

The research needs to be original in such a way that the thesis supervisor is convinced that this research has not been done before. The research also needs enough depth and still it must be possible to finish it in the set amount of time. It is possible for multiple students to cooperate in a research project as long as it is clear who did what. Moreover, every student has to write his own thesis reflecting his part of the research. The assessment will be based on the contents and form of the thesis and the presentation of this thesis.
Content aspects
The thesis describes the problem statement, research questions, approach and results of the research. This has to be done in a clear, structured and scientific manner. This includes:
• a clear introduction in which the problem statement and research questions are presented;
• the master’s student shows proper analysis of complex issues in a new context and is able to formulate a proper problem statement;
• a clear conclusion, based solely on the already used thought out principles and derived results;
• a clear line is shown between problem statements, approach, methods and the derived results;
• a motivation of the followed approach, reflecting on standard methods and their presuppositions,
• an adequate description of the followed approach;
• a purposeful and systematic way of collecting data;
• an honest, clear and concise description of the derived results, if necessary using tables;
• an analysis and discussion of the results;
• the usage of relevant and recent literature for the reasoning in the thesis.
• the correct usage of references.

Design aspects
Correct scientific references have to be used. Images and tables are accompanied by an index and caption. Mathematical formula, definitions, etc. have to be properly designed and numbered. The start and end of mathematical formulae have to be properly defined.

Language aspects
The thesis has to be written in English, considering correct spelling, syntactical structure of sentences and structure of content in paragraphs. The target audience consists of fellow master’s students and lecturers. Any jargon and/or abbreviations have to be explained unless they are common knowledge for this audience.

Citations
It is allowed to use several short citations. These citations have to be clearly referenced and have to be typographically distinguishable (that is, citations are placed in quotes). Non-allowed citations or missing references will result in a non-pass.
4 Facilities for Students

In this chapter, you will get an overview of the facilities that Maastricht University offers its students.

4.1 Student Affairs Office
The Student Affairs Office, among other things, takes care of the organization and administration of the education.

Visiting address: Paul-Henri Spaaklaan 1, 6229 GT Maastricht
Postal address: P.O. Box 616, 6200 MD Maastricht, the Netherlands.

Office hours:
PHS, C.1006 daily between 10h00 - 11h00 and 15h00 - 16h00.

Contact:
Admissions: dke-admissions@maastrichtuniversity.nl
Tel.: +31(0)43 388 33 59

Course booking and general questions: dke-studentaffairs@maastrichtuniversity.nl
Tel.: +31(0)43 388 35 25

Exams and graduation: dke-examination@maastrichtuniversity.nl
Tel.: +31(0)43 388 37 32

Scheduling: dke-scheduling@maastrichtuniversity.nl
Tel.: +31(0)43 388 59 17

Internationalization:
For any questions you may have about studying for a semester at a foreign university, or about a practical training abroad, for support, and for direct information you can contact DKE's international relations officer Wendy Brandt via:
dke-iro@maastrichtuniversity.nl.

4.2 Administrative structure of the Faculty

Administrative structure of the Faculty
The administrative structure of the Faculty is laid down in the faculty regulations.

The dean is responsible for the faculty's administration. More information is to be found on the website: www.maastrichtuniversity.nl/dke.

Faculty Board
The Faculty Board, chaired by the dean of the Faculty of Science and Engineering, runs the Faculty. The Faculty Board is charged with the general management and administration, as well as its policy regarding academic research and education.
Faculty Council
The Faculty Council is entitled to submit proposals and present their opinion to the Faculty Board regarding any matters relating to faculty administration, policy, education and research. The Faculty Council has rights of approval, e.g. regarding faculty regulations, research programmes, and the implementation of a binding study advice, and rights of advice, e.g. regarding the budget.

Directors of Studies
The directors of studies Dr. Pietro Bonizzi for the Bachelor programme and Dr. Matus Mihalak for the Master programmes are responsible for the organization and coordination of all teaching activities. The DKE Education Programme Committee (EPC) advises the directors of studies.

Education Programme Committee
Currently there is one EPC for the Bachelor's Data Science and Artificial Intelligence and the Master's Data Science for Decision Making and Artificial Intelligence. The EPC is responsible for advising the Faculty Board, the Director of Studies and the Board of Examiners. Furthermore, the EPC is entitled to advice in any subject related to the programme, and consists out of eight members, four students and four members of the academic staff. In addition, the quality assurance officer has the role of advisor in the committee.

All correspondence for the Education Programme Committee should be addressed to dke-secretariaat@maastrichtuniversity.nl or by postal mail to: Department of Data Science and Knowledge Engineering - Maastricht University P.O. Box 616, 6200 MD Maastricht.

Board of Examiners
The Board of Examiners is in charge of the organization and supervision of the examinations and is appointed by the Faculty Board.

All correspondence for the Board of Examiners should be addressed to dke-boe@maastrichtuniversity.nl.

Board of Admissions for the Master's Programmes
The Board of Admissions is responsible for granting the admission requests for entering a master's programme, and is appointed by the Faculty Board. All correspondence for the Board of Admissions should be addressed to dke-admissions@maastrichtuniversity.nl.

It is possible to follow incidental courses at the transnational University Limburg (located at Hasselt University, Belgium). Students who want to make use of this possibility should individually ask permission from the Board of Examiners of DKE, Maastricht University. More information on the transnational University Limburg, its staff members, and information on the content of the courses can be found at: www.uhasselt.be/informatica

4.3 Teaching Material
For each project, a project book will be published. The project books and the education schedules of each period are available two weeks before the start of a new period, at the latest. If the prospectus makes no mention of study material (= obligatory literature) or recommended literature, the study material will be mentioned on a separate book list that is available on the course information page in Canvas. All students will receive a communication from the study association Incognito.
4.4 Participation in the Education
The students are expected to be available from Monday through Friday from 08h30 to 18h00 for educational activities. Examinations can take place between 08h30 - 18h00 and exceptionally till 21h00.

4.5 Announcements concerning Educational Matters
Overviews and results of examinations will be published on the Student Portal. Please check your student email account daily for announcements or important messages from the university. This is the only way we can reach our students fast. Also use your student email account when contacting staff and fellow students. Always mention your student ID number in all correspondence.

4.6 Change of Address Student
Please check whether the information that you have entered in Studielink when you applied for our programme is still accurate. When we want to send you information or call you, we will refer to the information that has been entered when you registered. Adaptations to personal details can be made in Studielink, such as change of address, phonenumber and bank account details.

4.7 Project rooms
The rooms are open to students from Monday through Friday from 08h00 to 18h00. Wireless internet is available throughout the whole building. For questions on the systems, please refer to the system managers of DKE, tel. +31(0)43-388 54 93 or by mail: lo-fse@maastrichtuniversity.nl

House rules for all /project/meeting rooms:
• Users are not allowed to download illegally acquired materials;
• Users are not allowed to illegally download materials
• Users are not allowed to install illegally acquired software;
• Users should use their own devices for saving data, or save their data on their personal network drive (I:);
• Users should handle the furniture with care;
• It is strictly forbidden to eat or drink in these rooms;

4.8 Faculty Counsellors for Students
Study Adviser
The study advisers Tessa Fox and Wendy Brandt are staff members whom you can contact if you have any questions concerning your study. They can be reached at telephone number 043-3885361, in room C.1012 at PHS, or through dke-studyadvice@maastrichtuniversity.nl

They are familiar with the organization of the education, the faculty organization and the study. The study adviser is a primary advisor for students. If your study comes to a standstill, for whatever reason, you can contact the study adviser. It is also the right person to talk to if you have any questions to which you cannot find any answers in this student handbook or during faculty information meetings. But also in case of personal circumstances due to which your study or personal life are suffering, for instance illness or family circumstances, your study adviser can help. In a situation like that it is very important to contact the study adviser in time. The study adviser may also invite students for a talk if it appears that their results are falling back.
4.9 Student Services Centre (SSC)
The Student Services Centre is responsible for the preparation and execution of the policy of Maastricht University in the area of general student provisions. In short, this department has a number of specialized service units for student-related issues such as accommodation, sports, information on studies and work and career advice. In addition, there is a central information desk in the main entrance hall of the Visitors’ Centre, to which current and prospective students may address their questions. Visiting address: Bonnefantenstraat 2, Tel.: +31(0)43-388 53 88, www.maastrichtuniversity.nl/ssc.

4.9.1 Visitors’ Centre and student registration

Information Desk
The information desk in the UM Visitors’ Centre at Bonnefantenstraat 2 is the first point of contact for current and new students. It provides the following services:

- Help with admission and (re)registration;
- Information on and help with visas, scholarships, bank accounts and (health) insurance;
- Changing of address;
- Payment of tuition fees;
- Cancellation of registration;
- Reimbursement of tuition fees;
- Proof of payment/registration;
- Collection of your first UM-card;
- Help with housing;
- Appointments with student deans, psychologists, and career services;
- UM gifts.

Tel.: +31(0)43-388 53 88, e-mail: study@maastrichtuniversity.nl
Opening hours Monday to Friday, 08h30 to 18h00.

Visa and Scholarship Office
The Visa and Scholarship Office is responsible for immigration matters and scholarships for prospective and current students. For any questions on visas, please visit our website: www.maastrichtuniversity.nl/visa or e-mail: visa@maastrichtuniversity.nl. For any questions on scholarships, please visit the website: www.maastrichtuniversity.nl/scholarships.

4.9.2 UM Career Services
UM Career Services offers workshops, job interview simulations, quick career advice and more intensive career counseling. For more information, please see www.maastrichtuniversity.nl/careerservices or contact your study adviser at DKE

4.9.3 Student Guidance

Student Guidance
Psychological support (Student psychologists)
Student psychologists may be consulted in case of personal problems. Examples of complaints and problems include:

- Study related problems like study stress and fear of failure;
- Psychological complaints such as anxiety, depression, eating disorders, stress-related complaints, lack of confidence, dealing with traumatic experiences.
The student psychologists can help you by means of individual guidance and/or group training (in Dutch and English). Check the current offer here.
For making an appointment use the online tool on the website.
Study related legal support (Student deans).
For more information: www.maastrichtuniversity.nl/support
E-mail: studentendecanen@maastrichtuniversity.nl
Open during visiting hours at the SSC, please check the website for correct timeslots

**Studying with a disability, chronic illness or dyslexia**
It is important to Maastricht University that students with a functional impairment can successfully complete their studies without too much delay. By functional impairment UM means a psychological, cognitive, or physical impairment, either of a permanent or temporary character, that creates an inability to perform optimally. Amongst these are all motor, sensory or psychological disorders, but also non-visible disorders, such as dyslexia, chronic illness, physical complaints, depression and the like. Disability Support is available to students (with a functional impairment), prospective students, student counsellors, teachers, parents and others who are interested.

For more info: www.maastrichtuniversity.nl/disability
E-mail: disability@maastrichtuniversity.nl
Open visiting hours: Monday to Thursday from 11h00 to 13h00.
Tel: +31(0)43-388 52 72.

4.10 Education and Examination Regulations 2021-2022 (EER)

EER Bachelor 2021 - 2022
EER Master 2021 - 2022
Codes of conduct
5 Staff

5.1 Academic Staff

Dr. Stelios Asteriadis
Dr. Francesco Barile
Dr. Olivier Bilenne
Dr. Pietro Bonizzi
Dr. Martijn Boussé

Dr. Alexia Brassouli
Dr. Cameron Browne
Dr. Daniel Câmpora Pérez
Dr. Rachel Cavill
Dr. Steven Chaplick

Dr. Pieter Collins
Dr. Walter Crist III
Lucas Dahl
Dr. Otti D’Huys
Dr. Menica Dibenedetto
5.2 Lab assistants

Freek Oyen
Dean Boonen
5.3 Support Staff

Wendy Brandt  Esther Breuls  Céline Duijsens - Rondagh  Ivanka Dzon  Claire van Doorn

Tessa Fox  André Fraats  Charlotte Hamelers - Geelen  Iris Hoogsteder  Astrid Lamers

Anita Legtenberg  Bas Lemmens  Paola Möllering  Desirée Parren  Ellen Narinx - Schrauwen

Dénise van Spingelen  Miranda Vermeer  Dieudonnée van de Willige