Introduction to Data Science and Artificial Intelligence

Dept. of Advanced Computing Sciences

KEN1110

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
B. KhaertdinovA.M. WilbikP. Bonizzi

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full 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, 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 and artificial intelligence. As such, the course provides an overview of many topics that are addressed in much more detail throughout the Bachelor's Data Science and Artificial Intelligence programme.

Prerequisites

None.

Desired Prior Knowledge: The course appears as desired prior knowledge for the courses Reasoning Techniques and Theoretical Computer Science

Recommended reading

- S. Russell and P. Norvig (2010): Artificial Intelligence, A Modern Approach. Third edition, Pearson Education, ISBN 978-0-13-207148-2.
- C.D. Manning, P. Raghavan and H. Schütze (2008) Introduction to Information Retrieval. Cambridge University Press. ISBN 0521865719

Procedural Programming

Dept. of Advanced Computing Sciences

KEN1120

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
C. KouzinopoulosE. Hortal Quesada

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

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

Prerequisites

None.

It appears as part of the pre-requisites of the second semester project in year 1, both projects of year 2, the year 2 course Databases and the year 3 courses, Parallel Programming and Robotics and Embedded Systems.

The course appears as desired prior knowledge for the courses Introduction to Objects in Programming, Data Structures and Algorithms, Software Engineering, Databases and Machine Learning.

Recommended reading

H. Schildt, Java: A Beginner's Guide, Eighth Edition, ISBN: 1260440214, McGraw-Hill Education

Discrete Mathematics

Dept. of Advanced Computing Sciences

KEN1130

Period 1:

1 Sep 2025 **24 Oct 2025**

Credits:

4.0

Coordinator:

O. D'HuysM. Musegaas

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam

Full course description

In this course, we build a mathematical framework that is based on logic and reason. The main objective of the course is to make students familiar with the language of mathematics. Students will learn how to make sound arguments and to detect where and why certain arguments go wrong. For this purpose, we will discuss the basic principles of logic and, closely related, the basic types of mathematical proofs. In doing so, we will encounter numbers such as integers, natural numbers and real numbers and we shall examine what makes these numbers special. After that, we will use basic logic to discuss, among other things, the following mathematical concepts: infinity, sets, relations, functions, permutations and combinations. Our fundamental tool in all of this is plain common sense. You really do not need your toolbox of mathematical formulas learned in previous studies and neither do you need a calculator. Pen and paper are the basic instruments needed. After completing each topic, exercises will be provided to be completed in class or at home, since mathematics is mainly learned by practising repeatedly.

Prerequisites

None.

Recommended reading

Discrete Mathematics by A. Chetwynd and P. Diggle (ISBN: 9780340610473)

Project 1-1

Dept. of Advanced Computing Sciences

KEN1300

Semester 1:
1 Sep 2025 **23 Jan 2026**

Credits:
6.0

Coordinator:
M. Boussé

Teaching methods:
Skills, Work in subgroups, Presentation(s), Project-Centered Learning

Assessment methods:
Participation, Assignment, Presentation and paper

Full 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 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.
Project 1-1 will start in period 1.1 and period 1.2. The credits for the projects will become available at the end of period 1.3.
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.

Prerequisites

This project has no prerequisites. This project occurs as part of the prerequisites of project 2-1.

Objects in Programming

Dept. of Advanced Computing Sciences

KEN1220

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
D.J.N.J. SoemersT.H.J. Pepels

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course description

This course is a follow-up of the course Procedural Programming. 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.

Prerequisites

Desired prior knowledge: Basic Java Programming

Recommended reading

- C. Horstmann (2016). Java Concepts (8th Edition). John Wiley & Sons, New York, ISBN: 978-1-1190-5645-4
- C. Horstmann (2012). Big Java Late Objects. John Wiley & Sons, New York, ISBN 978-1-1180-8788-6

Calculus

Dept. of Advanced Computing Sciences

KEN1440

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
G.M. SchoenmakersM. Musegaas

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course description

The following subjects will be discussed in Calculus: limits and continuity, differential calculus, integral calculus, and an introduction to sequences and series and 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.

Prerequisites

None.

Recommended reading

University Calculus: Early Transcendentals, Global Edition (4th edition) by Joel Hass, Christopher Heil, Maurice D. Weir, and George B. Thomas (ISBN: 9781292317304).

Logic

Dept. of Advanced Computing Sciences

KEN1530

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
S.J. MaubachO. D'Huys

Teaching methods:
Lecture(s), Project-Centered Learning

Assessment methods:
Written exam

Full course description

This course covers classical logical systems as propositional logic and first-order predicate logic, and it presents an introduction to dynamic logic and hoare logic. The course covers notation systems, syntax and semantics, valid consequences, semantic tableaux and natural deduction proof systems

Prerequisites

None.

The course appears as a prerequisite for the course Logic for AI.

Recommended reading

"Logic in Action (Edition 2016) , Johan van Benthem, Hans van Ditmarsch, Jan van Eijck, Jan Jaspars" (www.logicinaction.org)

Linear Algebra

Dept. of Advanced Computing Sciences

KEN1410

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
P.W.L. DreesenM. Musegaas

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full 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.

Prerequisites

None.

Recommended reading

Linear Algebra and Its Applications (6th edition) by David C. Lay, Steven R. Lay, and Judi J. McDonald (ISBN: 978-1-292-35121-6).

Data Structures and Algorithms

Dept. of Advanced Computing Sciences

KEN1420

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
M.F.M. SondagF. Barile

Teaching methods:
Lecture(s), Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

As a continuation of the courses Procedural Programming and Objects in Programming, 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.

Prerequisites

Desired Prior Knowledge : Discrete Mathematics, Procedural Programming and Objects in Programming. The course is desired prior knowledge for Theoretical Computer Science.

The course itself occurs as part of the pre-requisites of both projects of year 2 and the third year course Parallel Programming.

Recommended reading

Required Reading: Sedgewick and Wayne (2011) Algorithms Fourth Edition. Addison Wesley. ISBN: 978-0321573513

Recommended Reading: A Y Bhargava (2016). Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People. Manning. ISBN: 978-1617292231

Principles of Data Science

Dept. of Advanced Computing Sciences

KEN1435

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
A. Wodeyar

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

Nowadays data science is at the core of modern society. We collect large amounts of data with the goal to make better decisions. We need to make sense of the data and leverage it in effective ways.

In this course, we will start with where data comes from—controlled experiments and observational studies. We will look at potential biases that can affect conclusions that we make from data. We will focus on what kind of causal statement one can draw based on data coming from experiments versus observational studies.

We will then summarize and visualize data using histograms and scatter plots. As we will see, there are some interesting recurring patterns when we summarize data. For example, the distribution of the average follows the bell shape curve. We will also consider deviations from the bell shape curve in case of outliers, and how to deal with real world and possible "unclean" data. Scatter plots will help us study the regression line and correlations.

This course will build the foundation for subsequent courses: probability and statistics, simulation and statistical analysis, and machine learning. You will learn how to convert data into tables and use them for subsequent analysis and plotting. We will focus on the principles of modern reproducible science, that is, to build analysis workflows that can easily be understood and re-run by others. We will learn how to keep track of analysis decisions and parameter choices. We will summarize all the uncertainties in an accessible way and see that this is crucial for effective decision making in the modern world.

During the labs, we will learn Python—one of the main programming languages used in data science—and how to use it to write analysis reports using literate programming—mixing code, plots, and narrative in the same document. We will analyze and visualize real datasets.

Prerequisites

Desired Prior Knowledge: Procedural Programming

Recommended reading

Study material: Statistics (fourth edition, 2007) by Freedman, Pisani, and Purves.
Additional selected material from data science textbooks and other resources.

Project 1-2

Dept. of Advanced Computing Sciences

KEN1600

Semester 2:
26 Jan 2026 **19 Jun 2026**

Credits:
6.0

Coordinator:
X. ZhangO. D'Huys

Teaching methods:
Skills, Work in subgroups, Presentation(s), Project-Centered Learning

Assessment methods:
Participation, Assignment, Presentation and paper

Full 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 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.

Project 1-2 will start in period 1.4 and period 1.5. The credits for the projects will become available at the end of period 1.6.

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.

Prerequisites

In order to participate in this project the student has to have passed **two out of four courses** from the set: Discrete Mathematics, Calculus, Procedural Programming and Objects in Programming.

Recommended reading

None.

Computational and Cognitive Neuroscience

Dept. of Advanced Computing Sciences

KEN1210

Period 5:

30 Mar 2026 **22 May 2026**

Credits:

4.0

Coordinator:

M. CapalboA.F. Roebroeck

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam

Full course 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.

Prerequisites

none

Recommended reading

Sternberg, R.J. (1999). Cognitive psychology (latest edition).

Fort Worth: Harcourt Brace.

Kalat, J.W. (2007) 9th edition Biological psychology. Pacific Grove, California; London: Brooks Cole.

Gazzaniga, M. (2009). Cognitive Neuroscience (third edition).

Software Engineering

Dept. of Advanced Computing Sciences

KEN1520

Period 5:

30 Mar 2026 **22 May 2026**

Credits:

4.0

Coordinator:

M. Pietrasik

Teaching methods:

Lecture(s)

Assessment methods:

Written exam, Assignment

Full course description

This course introduces students to software design and project management concepts. Students get 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 and principles, API development, code review, version control, specifications, debugging, refactoring, and abstract data types.

Prerequisites

Desired prior knowledge : Procedural Programming, Objects in Programming and Data Structures and Algorithms.

Recommended reading

- Goldman and Miller, MIT 6.031: Software Construction: http://web.mit.edu/6.031/
- Martin, Clean Code: A Handbook of Agile Software Craftsmanship (2008)

Numerical Methods

Dept. of Advanced Computing Sciences

KEN1540

Period 5:
30 Mar 2026 **22 May 2026**

Credits:
4.0

Coordinator:
B. SakçakM. Boussé

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course description

Numerical methods are techniques for solving problems from continuous mathematics (calculus and linear algebra) 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 methods for core problems of continuous mathematics, namely the solution of algebraic equations and differential equations, interpolating data by polynomials, numerically estimating derivatives and integrals, approximating functions by polynomials and trigonometric 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.

Prerequisites

Desired prior knowledge: calculus, linear algebra

Recommended reading

Recommended literature: J.D. Faires & R. Burden, "Numerical Methods", International 4th Edition, Cengage, 2012; ISBN: 978-0-495-38569-1.

Additional literature: C.F. Gerald & P.O. Wheatley, "Applied Numerical Analysis", Seventh Edition, Pearson, 2003; ISBN: 0-321-13304-8.
T. Siauw & A.M. Bayen, "An Introduction to Matlab Programming and Numerical Methods for Engineers", Academic Press, 2015; ISBN 978-0-12-520228-3.

Databases

Dept. of Advanced Computing Sciences

KEN2110

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
T.H.J. Pepels

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

This course will cover data modelling, the concepts and theory of the relational data model, and the widely used programming language SQL. These concepts will be applied in case studies.

Prerequisites

Desired Prior Knowledge : Procedural Programming and Objects in Programming.

Recommended reading

Ramakrishnan, R. and Gehrke, G. 2002. Database Management Systems (3 ed.). McGraw-Hill, Inc., New York, NY, USA.

Probability and Statistics

Dept. of Advanced Computing Sciences

KEN2130

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
C.J. Seiler

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

This course is a first introduction to probability and statistics. We will start by learning how to count and define a notion of probability. We will then move on to the concept of conditional probability, random variables and their distributions, expectation, continuous random variables, moments, joint distributions, and inequalities and limit theorems. This will provide us with the necessary language to study central topics of importance in statistics, such as the difference between a population and a sample, confidence intervals, parameter estimation, and hypothesis testing.

Prerequisites

None.

Desired prior knowledge : Discrete Mathematics and Calculus

Recommended reading

None.

Graph Theory

Dept. of Advanced Computing Sciences

KEN2220

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
M. Mihalak

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

A graph is simply a collection of objects and a binary relation between the objects. Visually, it is a bunch 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, colorings, and connectivity.

Prerequisites

Desired Prior Knowledge: Discrete Mathematics; Data Structures and Algorithms

Recommended reading

None.

Project 2-1

Dept. of Advanced Computing Sciences

KEN2300

Semester 1:
1 Sep 2025 **23 Jan 2026**

Credits:
6.0

Coordinator:
X. ZhangO. D'Huys

Teaching methods:
Skills, Work in subgroups, Presentation(s), Project-Centered Learning

Assessment methods:
Participation, Assignment, Presentation and paper

Full 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 2.1. 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.

Project 2-1 will start in period 2.1 and period 2.2 with weekly meetings. The credits for the projects will become available at the end of period 2.3.

Prerequisites

Students must have passed Project 1-1. Furthermore, the student has to have passed at least two out of the following three courses: Procedural Programming, Objects in Programming, and Data Structures and Algorithms.
The student furthermore needs to be registered for or has already completed at least three courses of the programme in year 2, semester 1.
This project is a prerequisite for Project 3-1.

Recommended reading

None.

Reasoning Techniques

Dept. of Advanced Computing Sciences

KEN2230

Period 2:

27 Oct 2025 **12 Dec 2025**

Credits:

4.0

Coordinator:

M.H.M. WinandsT.D. Rienstra

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam, Assignment

Full course 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.

Prerequisites

Desired Prior Knowledge: Introduction to Data Science and Knowledge Engineering, Logic.

Recommended reading

Required literature:

- Russell, S. and Norvig, P., Artificial Intelligence: A Modern Approach, 4th edition. Pearson, 2020.

- Bratko, I. (2012). Prolog: Programming for Artificial Intelligence, 4th edition. Addison-Wesley.

Machine Learning

Dept. of Advanced Computing Sciences

KEN2240

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
P. BosiljE.N. SmirnovE. Hortal Quesada

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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 and Clustering.
Lectures and practical assignments emphasize the practical use of the presented techniques and prepare students for developing real-world machine-learning applications.

Prerequisites

Desired prior knowledge : Procedural Programming, Calculus, Linear Algebra, Logic, Probability and Statistics.

Recommended reading

- T. Mitchell (1997). Machine Learning, McGraw-Hill, ISBN-13: 978-0071154673.
- H. Blockeel, Machine Learning and Inductive Inference (course text), Uitgeverij ACCO, 2012.
- I.H. Witten and E. Frank (2011). Data Mining: Practical Machine Learning Tools and Techniques (Third Edition), Morgan Kaufmann, January 2011, ISBN-13: 978-0123748560.

Simulation and Statistical Analysis

Dept. of Advanced Computing Sciences

KEN2530

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
J.M.H. Karel

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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.

Prerequisites

Desired Prior Knowledge : Knowledge: Probability & Statistics, Calculus, Matlab, and Java.

Recommended reading

Object-Oriented Computer Simulation of discrete-event systems – Jerzy Tyszer,
Design and Analysis of Experiments – Douglas C. Montgomery,
Introduction to Probability Models – Sheldon M. Ross.

Human Computer Interaction & Affective Computing

Dept. of Advanced Computing Sciences

KEN2410

Period 4:

26 Jan 2026 **27 Mar 2026**

Credits:

4.0

Coordinator:

K. ZarkogianniY.C. Semerci

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam

Full 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.

Prerequisites

Desired Prior Knowledge: Machine Learning, Probabilities and Statistics.

Recommended reading

- Shneiderman B, Plaisant C, Cohen M, Jacobs S, Elmqvist N, Diakopoulos N. (2016) Designing the user interface: strategies for effective human-computer interaction. Pearson, ISBN: 978-0134380384
- Calvo RA, D'Mello S, Gratch JM, Kappas A, (2015). The Oxford handbook of affective computing. Oxford University Press, ISBN: 978-0199942237

- Coursera video lectures of Scott Klemmer and accompanying slides.

Mathematical Modelling

Dept. of Advanced Computing Sciences

KEN2430

Period 4:

26 Jan 2026 **27 Mar 2026**

Credits:

4.0

Coordinator:

R.L.M. PeetersJ.M.H. Karel

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam, Assignment

Full course 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.

Prerequisites

Desired Prior knowledge : Linear Algebra, Calculus, Matlab.

Recommended reading

Richard J. Vaccaro, Digital Control: A State-Space Approach, McGraw-Hill, 1995, ISBN 0-07-066781-0.

Natural Language Processing

Dept. of Advanced Computing Sciences

KEN2570

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
A.S. HärmäG. Spanakis

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

ChatGPT can answer almost any question you have. Siri can tell me when I need an umbrella. But how do they work? Over the past few years, 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 to 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.

Prerequisites

None.

Desired Prior Knowledge : Procedural Programming and Objects in Programming, Probability and Statistics, Machine Learning

Recommended reading

1. Daniel Jurafsky and James H. Martin. "Speech and language processing an introduction to natural language processing, computational linguistics, and speech." Pearson, London, 2000
2. Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA. 1999

Project 2-2

Dept. of Advanced Computing Sciences

KEN2600

Semester 2:
26 Jan 2026 **19 Jun 2026**

Credits:
6.0

Coordinator:
K. Schneider

Teaching methods:
Work in subgroups, Project-Centered Learning

Assessment methods:
Participation, Assignment, Presentation and paper

Full course description

In this project, students individually indicate their preferences for a given umbrella project. A least regret algorithm ensures that students will be allocated to a group of 6-7 students corresponding to an umbrella project such that the overall regret is minimal. In their project groups, the students come up with a project topic fitting to their umbrella topic in co-creation with their tutor and the examiners.

The project is divided into three phases. In the first phase, the groups define the topic and the project scope. At the end of that phase, they hand in a project plan and a prototype. The second phase is the implementation phase. Students should aim for a first version of the product that already allows for meaningful experiments. At the end of that phase, the students meet with the examiners in a formative assessment moment to discuss their progress and future plans. The third phase is used to enhance the product, write the project report and to create a poster that will be presented in a poster fair at the end of the project. The students' understanding of the project is tested in a project defense.

During the project phases, each group is guided by a tutor. The students work part-time in phases 1 and 2 of the project (with a full-time project week in-between), while they work full-time on the project in phase 3.

Prerequisites

Students must have passed Project 1-2. Furthermore, the student has to have passed at least two out of the following three courses: Procedural Programming, Objects in Programming, and Data Structures and Algorithms. The student furthermore needs to be registered for or has already completed at least three courses of the programme in year 2, semester 2. This project is not a prerequisite for another project / course.

Recommended reading

None.

Philosophy & Artificial Intelligence

Dept. of Advanced Computing Sciences

KEN2120

Period 5:

30 Mar 2026 **22 May 2026**

Credits:

4.0

Coordinator:

C.J.M.E. JuhaszR. Gianni

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam

Full course 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.

Prerequisites

None.

Recommended reading

None

Linear Programming

Dept. of Advanced Computing Sciences

KEN2520

Period 5:

30 Mar 2026 **22 May 2026**

Credits:

4.0

Coordinator:

S.M. Kelk

Teaching methods:

Project-Centered Learning

Assessment methods:

Written exam, Assignment

Full course 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.

Prerequisites

Desired Prior Knowledge: Linear Algebra.

Recommended reading

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

DACS Honours Programme - KE@Work (2-1)

Dept. of Advanced Computing Sciences

KEN2310

Semester 1:

1 Sep 2025 **23 Jan 2026**

Credits:

6.0

Coordinator:

K. Driessens

Teaching methods:

Assignment(s), Working visit(s)

Assessment methods:

Final paper

Full course description

The DACS Honours Programme consists KnowledgeEngineering@Work (KE@Work) and (MaRBLe 2.0). Students admitted to the KE@Work path are placed at a company or organization in the region through a careful selection and matching process. During the full second and third year of the bachelor's programme, 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 DACS supervisors

DACS Honours Programme - MaRBLe 2.0 (2-1)

Dept. of Advanced Computing Sciences

KEN2320

Semester 1:
1 Sep 2025 **23 Jan 2026**

Credits:
6.0

Coordinator:
R. Cavill

Teaching methods:
Assignment(s), Working visit(s)

Assessment methods:
Final paper

Full course description

During 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. Participation is open to excellent and motivated students.

Prerequisites

None.

Recommended reading

None.

DACS Honours Programme - KE@Work (2-2)

Dept. of Advanced Computing Sciences

KEN2610

Semester 2:

26 Jan 2026 **19 Jun 2026**

Credits:

6.0

Coordinator:

K. Driessens

Teaching methods:

Assignment(s), Working visit(s)

Assessment methods:

Final paper

Full course description

The DACS Honours Programme consists KnowledgeEngineering@Work (KE@Work) and (MaRBLe 2.0). Students admitted to the KE@Work path are placed at a company or organization in the region through a careful selection and matching process. During the full second and third year of the bachelor's programme, 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 DACS supervisors.

Prerequisites

None.

Recommended reading

None.

DACS Honours Programme - MaRBLe 2.0 (2-2)

Dept. of Advanced Computing Sciences

KEN2620

Semester 2:
26 Jan 2026 **19 Jun 2026**

Credits:
6.0

Coordinator:
R. Cavill

Teaching methods:
Assignment(s), Working visit(s)

Assessment methods:
Final paper

Full course description

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. MaRBle takes place in year two of the bachelor's programme.

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.

Recommended reading

None.

Game Theory

Dept. of Advanced Computing Sciences

KEN2580

Period 5:
30 Mar 2026 **22 May 2026**

Credits:
4.0

Coordinator:
M. Salvioli

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Discrete Mathematics, Linear Algebra

Recommended reading

None

Introduction to Image & Video Processing

Dept. of Advanced Computing Sciences

KEN3238

Period 5:
30 Mar 2026 **22 May 2026**

Credits:
4.0

Coordinator:
G. GoyalY.C. Semerci

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Presentation, Assignment

Full course description

In this course, students will be introduced to the core principles and techniques of image and video processing, emphasizing how visual information is represented, manipulated, and interpreted by computational systems. Students will learn how images and videos can be expressed as numerical data and how this representation enables a variety of processing techniques. The course covers a wide range of topics, organized around six core concepts: representation, transformation, enhancement, compression, recognition, and generation. The topics include understanding image and video formats and color models, performing geometric and intensity-based transformations, designing and applying filters, restoring degraded images, video stabilization, compressing images and videos for efficient storage, identifying and isolating features or objects, and synthesizing new images and videos through computational methods. By the end of the course, students will be equipped to apply key image and video processing techniques and understand how they contribute to larger pipelines in visual analysis and interpretation.

Prerequisites

Desired Prior-knowledge: Probability and Statistics, Machine Learning, Python

Recommended reading

None

Bachelor's thesis

Dept. of Advanced Computing Sciences

KEN3500

Year:
1 Sep 2025 **19 Jun 2026**

Credits:
18.0

Coordinator:
E. Hortal Quesada

Teaching methods:
Paper(s)

Assessment methods:
Presentation and paper

Full course description

At the end of the Bachelor’s study in Data Science and Computing Sciences 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.

Prerequisites

In order to start working on the thesis, a student needs to have obtained at least 138 ECTS (among which are 60 credits of the first year, and 40 ECTS of the second year).

For the complete procedure and relevant forms see https://intranet.maastrichtuniversity.nl/en/dke-students/thesis

Operations Research

Dept. of Advanced Computing Sciences

KEN3410

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
N. YousefimaneshS.M. Kelk

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course 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 queuing systems, Markov chains and Markov decision problems. As background students will be introduced to the methodological similarities and differences between OR and data science.

Prerequisites

None.

Recommended reading

None.

Intelligent Systems

Dept. of Advanced Computing Sciences

KEN3430

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
K. Driessens

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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.

Prerequisites

None.

Recommended reading

None.

Data Analysis

Dept. of Advanced Computing Sciences

KEN3450

Period 4:
26 Jan 2026 **27 Mar 2026**

Credits:
4.0

Coordinator:
M.F.M. SondagG. Spanakis

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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.).

Prerequisites

None.

Desired prior knowledge: Simulation and Statistical Analysis.

Recommended reading

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

Digital Society

Dept. of Advanced Computing Sciences

KEN3111

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
M.B. Archer

Teaching methods:
Project-Centered Learning

Assessment methods:
Final paper, Participation, Presentation, Oral exam

Full course description

Digital technologies affect our lives in increasingly profound and inescapable ways. Digitalization has changed the way we interact with friends, family members, and romantic partners. It has changed the way we access information and, subsequently, the kind of information we are able to access. Digital technologies have impacted trust in institutions and have precipitated new societal expectations regarding the relationship between governments and constituents. At the same time, it is important not to attribute these changes solely to the introduction of new technologies, but to instead understand these technologies as situated in specific social, economic, and political contexts.

The phrase ‘digital society’ attunes us to the interplay between the digital and the social, challenging the boundary between these two domains, similar to the way the notion of socio-ecology challenges the boundary between society and the environment. The goal of this 4 ECTS course is to equip students with a conceptual vocabulary to critically analyze the impacts of digitalization on and within society and the environment, and vice versa. Through a combination of lectures and discussion seminars, we will critically examine some of these entanglements through a selection of case studies and examples, from e-waste to ChatGPT to surveillance technologies.

Learning objectives

1. Knowledge and understanding: Supported by advanced texts, students develop a systematic understanding of new insights and methods in the academic field studying the relation between

digital technology and social aspects, the digital economy, privacy, surveillance, fakeness and socio-ethics.
2. Applying knowledge and understanding: Students can devise and sustain academic arguments pertaining to the relation between digitalization and society. Capable of conceiving, designing, and conducting a substantial process of research on platformization and datafication.
3. Making informed judgments: Students will be able to gather and interpret relevant data with the aim to further sharpen their ability to critically analyze, evaluate, and synthesize new and complex ideas from the academic field studying the relation between digital technology and social conditions, the digital economy, privacy, surveillance, fakeness, and socio-ethics.
4. Communication: Students fortify their capacity to communicate ideas, problems, and solutions pertaining to (the effects of) digitalization, datafication, platformisation in society with (a) peers, (b) the larger scholarly community, (c) non-IT- experts and (d) society in general. They develop a sense of multi-disciplinary group-work and learn to communicate and collaborate across disciplinary borders.
5. Learning skills: This course hands students the contexts, tools, insights, and techniques to help them to continue their studies of the complex relations between socio-technical developments (e.g. digitalization, datafication, platformisation) and societal outcomes (e.g. disintermediation, data colonialism, ethics) independently and autonomously.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

This is a shared course (maximum capacity of 60 students).

Prerequisites

None.

Recommended reading

None.

Game Theory

Dept. of Advanced Computing Sciences

KEN3130

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
G.M. Schoenmakers

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Discrete Mathematics, Linear Algebra

Recommended reading

None.

Semantic Web

Dept. of Advanced Computing Sciences

KEN3140

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
R. Celebi

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Participation, Assignment

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Desired Prior Knowledge: Logic.

Recommended reading

The documents on the site of the World Wide Web Consortium (W3C).

Multi-Scale Modelling of Biological Syst

Dept. of Advanced Computing Sciences

KEN3170

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
M. BreuerM. Summer - Kutmon

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

This course introduces computational approaches for modelling biological systems across different scales, from molecular to population-level processes, using a range of mathematical and computational frameworks. Students will apply concepts from calculus, statistics, and mathematical modelling to real-world biological problems, leveraging techniques such as ordinary differential equations (ODEs), constrained-based modelling, and graph theory. The course emphasizes hands-on application of models to explore molecular interactions, tissue dynamics, physiology, and population-level phenomena.

The course is structured to progress through biological scales over six weeks. Week one introduces constrained-based modelling of genome-scale metabolic models using Python and COBRApy, with omics data integration to refine models. This is followed by exploring gene regulatory networks building on graph theory concepts and Boolean models implemented in Python or R and Cytoscape for visualizing dynamic network behaviour. Weeks 3 and 4 transition to the tissue level, where students will use mechanistic models to examine tissue dynamics. These weeks use MATLAB with an emphasis on plant systems and neuroscience as case studies. Week five covers population-level phenomena like ODE-based epidemiological models to simulate the spread of diseases in Phyton. Finally, the last week brings together insights from previous weeks, addressing the challenges and strategies for integrating multi-scale biological data into computational models. Throughout the course, we will emphasize the use of tools like Github and Jupyter Notebooks for collaborative programming, reinforcing programming skills in different relevant programming languages.

Prerequisites

None.

Desired Prior Knowledge : Procedural Programming, Calculus, Linear Algebra, Bio-informatics.

Recommended reading

None

Robotics and Embedded Systems

Dept. of Advanced Computing Sciences

KEN3236

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
B. SakçakR. Möckel

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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 DACS. 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Maximum number of 130 students can follow this course. (Shared with the other bachelor)

Prerequisites

Procedural Programming and Objects in Programming.

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

Recommended reading

None.

Introduction to Quantum Computing

Dept. of Advanced Computing Sciences

KEN3241

Period 1:
1 Sep 2025 **24 Oct 2025**

Credits:
4.0

Coordinator:
D. Dibenedetto

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course 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:

1. most common models of quantum computation (e.g., quantum circuits and measurement-based quantum computing).
2. 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.
3. 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Linear Algebra.

Recommended reading

- Isaac Chuang, Michael Nielsen, “Quantum Computation and Quantum Information”, 10th Anniversary Edition, Cambridge University Press, 2011.
- N. David Mermin, “Quantum Computer Science: An Introduction”, 1st Edition, Cambridge University Press, 2007

Project 3-1

Dept. of Advanced Computing Sciences

KEN3300

Semester 1:
1 Sep 2025 **23 Jan 2026**

Credits:
6.0

Coordinator:
R. MöckelK. Schneider

Teaching methods:
Work in subgroups, Project-Centered Learning

Assessment methods:
Participation, Assignment, Presentation and paper

Full course description

The students in project 3-1 will work on project tasks of diverse fitting to their curriculum provided by (external) customers. The project will be carried out by small groups of students, which will be guided by a tutor. Company supervisors or staff members of the Department of Advanced Computing Sciences act as customers of the projects. Additionally, two examiners of the Department of Advanced Computing Sciences assess the students' achievements. The group composition will be based on the students' preferences.

The project is split up into three phases coinciding with the three periods of the first semester in year 3:

In the first phase, students will become familiar with the project assignment, and lay a foundation for the next phases by scoping the project well. The students meet their customers and discuss the framework, requirements, and deliverables of the project. Based on the outcome of this discussion and on research on the project topic, the students create a project plan including, e.g., a risk analysis, a literature review, and a time planning. Last but not least, a prototype needs to be submitted at the end of phase 1.

The aim of phase 2 is to implement a solid product that can further be enhanced in phase 3. More advanced concepts will be implemented and a planning for phase three is made. In the midway evaluation at the end of phase 2, the experimental results should help the students to convince their customers that the project can successfully be completed in phase 3.

Phase 3 consists of three full-time weeks, in which the students improve their product based on their ambitions and agreements with the customer. They have to document their work in the form of a scientific report and a business presentation. Furthermore, the students have to defend their choices and implementations in a project defense with the DACS examiners.

The project will be accompanied by so-called skills classes. These classes help the student develop competencies that are useful in this project and their further careers.

Project 3-1 will start in period 3.1 and period 3.2 with weekly meetings. The credits for the projects will become available at the end of period 3.3.

Prerequisites

Project 2-1.

Recommended reading

None.

DACS Honours Programme - KE@Work (3-1)

Dept. of Advanced Computing Sciences

KEN3310

Semester 1:
1 Sep 2025 **23 Jan 2026**

Credits:
6.0

Coordinator:
K. Driessens

Teaching methods:
Assignment(s), Working visit(s)

Assessment methods:
Final paper

Full course description

The DACS Honours Programme consists KnowledgeEngineering@Work (KE@Work) and (MaRBLe 2.0). Students admitted to the KE@Work path are placed at a company or organization in the region through a careful selection and matching process. During the full second and third year of the bachelor's programme, 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 DACS supervisors.

Prerequisites

None.

Recommended reading

None.

Study Abroad

Dept. of Advanced Computing Sciences

KEN3600

Semester 1:
1 Sep 2025 **23 Jan 2026**
Semester 2:
26 Jan 2026 **19 Jun 2026**

Credits:
30.0

Coordinator:
M. Musegaas

Assessment methods:
Written exam, Attendance, Assignment

Full course description

DACS offers its students the possibility to study a semester abroad at one of DACS partner universities. Third year bachelor's students and 2nd year master's students can get the opportunity to study a semester abroad, as part of their education programme in Maastricht. The credits received abroad will be transferred / part of your programme at DACS in Maastricht.Of course, this is only possible after approval of the Board of Examiners. There are several universities where DACS can send its students to.

If you still have questions afterwards you may contact our Exchange Coordinator Luc Giezenaar via: dacs-iro@maastrichtuniversity.nl

Prerequisites

You have to obtained at least 40 ECTS of year 1 courses.

Computer Security

Dept. of Advanced Computing Sciences

KEN2560

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
M.E. Glazunov

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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 (basic concepts, access control, security policies and mechanisms, cryptography, cryptoanalysis, protocols), 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 obligatory assignments and an obligatory team project where the students will have to solve some of the most important issues we discussed in classroom. Finally, the course concludes with a closed book exam.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Desired Prior Knowledge : Procedural Programming, Objects in Programming, Data Structures and Algorithms, Software Engineering, Databases.

Recommended reading

1. Pfleeger, C. P., Pfleeger, S. L., & Margulies, J. (2023). Security in computing (5th ed.). Pearson.

2. Bishop, M. (2018). Computer security: Art and science (2nd ed.). Addison-Wesley Professional.

3. Maymi, F., & Harris, S. (2021). CISSP all-in-one exam guide (9th ed.). McGraw Hill.

4. Buchmann, J. (2004). Introduction to Cryptography (2nd ed.). Springer.

Recommender Systems

Dept. of Advanced Computing Sciences

KEN3160

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
N. TintarevF. Barile

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course description

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 recommender systems, Contextual Recommender systems

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Machine Learning.

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

Recommended reading

Jannach, Dietmar, et al. *Recommender systems: an introduction.* Cambridge University Press, 2010.
Additional research papers and online articles

Logic for Artificial Intelligence

Dept. of Advanced Computing Sciences

KEN3231

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
T.D. Rienstra

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

Logic

Recommended reading

A syllabus and scientific literature.

Parallel Programming

Dept. of Advanced Computing Sciences

KEN3235

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
C. TerbovenB. Küppers

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2

Maximum number of 50 students can follow this course.

Prerequisites

Procedural Programming ,Objects in Programming, Data Structures and Algorithms.

Recommended reading

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

Large Scale IT and Cloud Computing

Dept. of Advanced Computing Sciences

KEN3239

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
M. PolitzeT.T. Eifert

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

A maximum number of 120 students can take this course. (shared course with the other bachelor)

Prerequisites

None.

Recommended reading

None.

Business Process & Knowledge Management

Dept. of Advanced Computing Sciences

KEN3242

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
P.W.L. Bollen

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam

Full course description

The course Business Process & Knowledge Management offers a comprehensive overview of the core topics in Business process management and Knowledge management, both from a conceptual business perspective. Particular emphasis is put on the basic techniques and methods, and some example application domains of computer science, data science and artificial intelligence

Business Process management is the practice of managing and optimizing the processes that are used to complete tasks and achieve goals. *Knowledge Management* studies how organizations can exploit, manage and retain their knowledge resources

The educational goal of this block is to provide the students of Data Science & Artificial Intelligence and Computer Science bachelor with an integrated perspective on business process and knowledge management. We will provide students with the basic processes for capturing the elementary business process characteristics in terms of events, activities, tasks, gateways and how they link to the types of data that provide inputs and outputs to these processes by means of business process models. We will apply the concepts on practical examples from the professional practical for computer and data science graduates.

Prerequisites

None.

Recommended reading

None

Introduction to Bio-Informatics

Dept. of Advanced Computing Sciences

KEN3440

Period 2:
27 Oct 2025 **12 Dec 2025**

Credits:
4.0

Coordinator:
R. Cavill

Teaching methods:
Project-Centered Learning

Assessment methods:
Written exam, Assignment

Full course 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.

This is an optional course: Third year students choose three electives per period out of the optional courses during period 1 and 2.

Prerequisites

None.

Desired Prior Knowledge: Procedural Programming, MatLab.

Recommended reading

Introduction to Computational Genomics, A Case Studies Approach, Nello Cristianini, Matthew W. Hahn, Cambridge University Press, 2006, Hardback and Paperback (ISBN-13: 9780521856034 | ISBN-10: 0521856035).