CONSTRUCTOR
UNIVERSITY

Study Program Handbook

Applied Computer Science

Bachelor of Science

Subject-specific Examination Regulations for Applied Computer Science (Fachspezifische Prüfungsordnung)

The subject-specific examination regulations for Applied Computer Science are defined by this program handbook and are valid only in combination with the General Examination Regulations for Undergraduate degree programs (General Examination Regulations = Rahmenprüfungsordnung). This handbook also contains the program-specific Study and Examination Plan (Chapter 5).

Upon graduation, students in this program will receive a Bachelor of Science (BSc) degree with a scope of 180 ECTS (for specifics see Chapter 3 of this handbook).

Version	Valid as of	Decision	Details
Fall 2025 - V1	Sept. 1 2025		
		Feb 22, 2023	Academic Senate approval of study program name change from “Computer Science and Software Engineering” to “Applied Computer Science”
		Jan 25, 2023	Originally approved by Academic Senate

Contents

1 Program Overview

1.1 Concept

1.1.1 Constructor University Educational Concept

Constructor University aims to educate students for both an academic and a professional career by emphasizing three core objectives: academic excellence, personal development and employability to succeed in the working world. Constructor University offers an excellent-research driven education experience across disciplines to prepare students for graduate education as well as career success by combining disciplinary depth and interdisciplinary breadth with supplemental skills education and extra-curricular elements. Through a multi-disciplinary, holistic approach and exposure to cutting-edge technologies and challenges, Constructor University develops and enables the academic excellence, intellectual competences, societal engagement, professional and scientific skills of tomorrows leaders for a sustainable and peaceful future.

In this context, it is Constructor University's aim to educate talented young people from all over the world, regardless of nationality, religion, and material circumstances, to become citizens of the world who are able to take responsible roles for the democratic, peaceful, and sustainable development of the societies in which they live. This is achieved through a high-quality teaching as well as manageable study loads and supportive study conditions. Study programs and related study abroad programs convey academic knowledge as well as the ability to interact positively with other individuals and groups in culturally diverse environments. The ability to succeed in the working world is a core objective for all study programs at Constructor University, both in terms of actual disciplinary subject matter and also to the social skills and intercultural competence. Study-program-specific modules and additional specializations provide the necessary depth, interdisciplinary offerings provide breadth while the university-wide general foundation and methods modules, optional German language and Humanity modules, and an extended internship period strengthen the employability of students. The concept of living and learning together on an international campus with many cultural and social activities supplements students' education. In addition, Constructor University offers professional advising and counseling.

Constructor University's educational concept is highly regarded both nationally and internationally. While the university has consistently achieved top marks over the last decade in Germany's most comprehensive and detailed university ranking by the Center for Higher Education (CHE), it has also been listed by one of the most widely observed university rankings, the Times Higher Education (THE) ranking. More details on the current ranking positions can be found at https://constructor.university/more/about-us.

1.1.2 Program Concept

Digitalization is a key driver of innovation and success across all industries. Applied Computer Science is obviously a key element in these processes. At the same time, there is a substantial change in the way daily work is organized and carried out. The share of home office and remote work increases, e.g., to collaborate with team members who are distributed around the world or to control, monitor, and maintain facilities and processes from a distance. While offering a lot of opportunities in terms of convenience for employees and reduced costs for employers, this new normal of working also requires different skills and knowledge of the related tools and methods, which are addressed by this program.

Furthermore, online education is changing the higher education landscape in profound ways. It caters for specific needs and interests of students, especially in terms of the flexibility in which they can carry out their studies. And it is a natural option to prepare them for the new normal of remote work.

The bachelor program in Applied Computer Science uses online education with high amounts of flipped-classroom elements. This means that students participate in online courses with predominantly asynchronous lectures and exercise material, which are complemented by tutorials and hands-on sessions. Students are guided and supported by faculty as well as experienced tutors and lecturers to transfer the acquired knowledge into practice. The hands-on elements include high amounts of collaboration with other students, use of tools and concepts to engage in distributed work from different places in potentially different time-zones, and remote access to physical devices and set-ups.

The Computer Science core of the program is complemented with Management and Leadership modules in the second and third study years. Students will not only be trained in programming and software development, but will also acquire fundamental knowledge in business and learn how innovations can be transferred into a marketable product. Furthermore, they may take part in interdisciplinary courses in which problems are tackled from a wider perspective challenging them to think outside the boundaries of their discipline.

Overall, by completing their studies, students will be able to directly enter the job market or to continue their studies in a graduate program, for example the MSc in Computer Science and Software Engineering offered at Constructor University. Apart from the solid knowledge and skills obtained in Applied Computer Science, graduates are particularly well prepared for the demands of modern work, i.e. to work remotely and as part of a diverse team.

1.2 Specific Advantages of Applied Computer Science at Constructor University

The Applied Computer Science program at Constructor University aims to provide an application-oriented knowledge of Computer Science including a preparation for important aspects of modern professional life, namely remote work and life-long learning.

The educational approach of the faculty is to relate the theoretical contents of the discipline to their contemporary application in industry and research. The instructors aim to include recent developments of the topics covered to demonstrate how basic methods or techniques are applied today and how the material covered relates to the challenges of digitalization and the related state of the art in research and development.

- Early involvement in software development project work is an essential aspect of the study program which further extends the already positively acknowledged educational approach in Computer Science at Constructor University.
- The Computer Science faculty’s pedagogy, together with the positive teaching environment, has been acknowledged in several rankings: In the Computer Science ranking published by the Centre for Higher Education (CHE) in 2015, the support by instructors and the relationship to research were ranked 1st of 68 study programs. In the European U-Multirank ranking published

in 2018, the overall learning experience in Computer Science was ranked 10th and research-oriented teaching in Computer Science was ranked 2nd of 304 European universities offering Computer Science programs.

- The involvement of students and alumni in the program development process using a direct and open dialogue is going to ensure that the program will be constantly fine-tuned to the specific needs of students, such as covering certain topics at a certain time with respect to the preparation of internship or job applications.
- Computer Science student teams participate regularly in international programming competitions. Constructor University hosted the Northwestern European Regional Contest (NWERC) of the ACM International Collegiate Programming Contest on campus in 2010 and 2011. Student teams participate in NWERC competitions since then on an annual basis. In 2014, students organized the first JacobsHack! hackathon on campus, which was sponsored, among others, by Google, Microsoft, and SAP. The 2018 edition of JacobsHack!, sponsored, among others, by Facebook, Skyscanner, GitHub and Bloomberg, attracted participants from all over Europe. As the program features important elements remote collaborative software development, there is also the option for online students to participate in according activities if they are interested in them.

1.3 Program-specific Educational Aims

1.3.1 Qualification Aims

The program is an online program with optional blended elements, e.g., in summer. Lectures incorporate asynchronous material and primarily follow a flipped classroom model, i.e., including application components in the spirit of problem-based- as well as project-based-learning. Practical components, particularly labs, projects, and thesis are based on remote access, distributed development. Tutoring includes virtual study groups, peer evaluation and mentoring by faculty. Performance evaluation are conducted as online e-exams.

The remote work aspects include collaborative software development and remote access to physical devices for, e.g., control, monitoring and maintenance. Due to the aspects of independent, self-governed knowledge acquisition, the students are prepared for life-long learning, where additional knowledge and skills need to be acquired or updated in a regular fashion, especially in fast moving areas like Computer Science.

The main subject-specific qualification aim is to enable students to take up qualified employment in modern industries involving digitalization and information technology or to enter graduate programs related to Applied Computer Science. Graduates of the Applied Computer Science program have obtained the following competencies:

- Applied Computer Science competence
  Graduates are familiar with the foundations of Computer Science and they are able to design and develop software addressing a given application scenario. They are able to analyze and structure complex problems and they are able to address them using methods of Applied Computer Science. Graduates are able to construct and maintain complex computer systems using a structured, analytic, and creative approach. They are trained in developing software in collaborative teams in a remote fashion, i.e., independent of the location they live and work at.

- Communication competence
  Graduates are able to communicate subject-specific topics convincingly in both spoken and written form to fellow computer scientists or to customers.

- Teamwork and project management competence
  Graduates are able to work effectively in a remote team and they are able to organize workflows in complex development efforts. They are familiar with tools that support the development, testing, and maintenance of large software systems and they are able to take design decisions in a constructive way.

- Learning competence
  Graduates have acquired a solid foundation enabling them to assess their own knowledge and skills, learn effectively, and remain up-to-date with the latest developments in the rapidly evolving field of Applied Computer Science.

- Personal and professional competence
  Graduates are able to develop a professional profile, justify professional decisions based on theoretical and methodical knowledge, and critically reflect on their behavior with respect to their consequences for society.

- Management competence
  Graduates have obtained basic business and management knowledge supporting them to reflect their core discipline against the background of a corporate environment and to incorporate a business perspective into computer science and software development.

1.3.2 Intended Learning Outcomes

By the end of the program, students will be able to:

1. acquire Applied Computer Science knowledge in an independent, self-governed way;
2. work in teams distributed around the globe to analyze complex problems, to evaluate them, and to derive solutions;
3. comprehend the processes and tools of Software Engineering for collaborative, remote software and systems development;
4. program software in C/C++ and understand algorithms;
5. be able to use libraries and to generate software in core Computer Science areas;
6. apply suited mathematical methods;
7. understand operating systems, databases, and web services;
8. comprehend methods from Artificial Intelligence and Machine Learning;
9. understand the relation between software and its links to the physical world;
10. analyze data and to extract insights from it;
11. apply the acquired Software Engineering skills and Computer Science knowledge in collaborative, remote projects;
12. use academic or scientific methods as appropriate in the field of Applied Computer Science such as defining research questions, justifying methods, collecting, assessing and interpreting relevant information, and drawing scientifically-founded conclusions that consider social, scientific and ethical insights;
13. develop and advance solutions to problems and arguments in their subject area and defend these in discussions with specialists and non-specialists;

14. engage ethically with academic, professional and wider communities and to actively contribute to a sustainable future, reflecting and respecting different views;
15. take responsibility for their own learning, personal and professional development and role in society, evaluating critical feedback and self-analysis;
16. apply their knowledge and understanding to a professional context;
17. take on responsibility in a diverse team;
18. adhere to and defend ethical, scientific and professional standards.

1.4 Career Options and Support

Digitalization is affecting all areas of business, industry, daily life, and society. There is accordingly a very high demand for graduates with a background in Applied Computer Science in general. In addition, students have been trained to be able to work in a remote, collaborative fashion and being able to engage in life-long learning, i.e., to acquire or update knowledge and skills in the fast-moving areas of Computer Science in an independent and self-governed way. This offers not only increased flexibility for graduates to engage in professional opportunities worldwide, it is also a substantial benefit for potential employers as they may select from an increased pool of talented candidates, whom they do not need to relocate to work on their job.

The areas of employment are almost unlimited as digitalization is important in business, industry, daily life, and society. Within these areas, research & development or management tracks can be taken. The job market includes jobs such as software engineer, information systems manager, data analyst, computer systems engineer, application developer, IT consultant, remote maintenance manager, and system analyst.

The Career Service Center (CSC) helps students in their career development. It provides students with high-quality training and coaching in CV creation, cover letter formulation, interview preparation, effective presenting, business etiquette, and employer research as well as in many other aspects, thus helping students identify and follow up on rewarding careers after graduating from Constructor University. Furthermore, the Alumni Office helps students establish a long-lasting and global network which is useful when exploring job options in academia, industry, and elsewhere.

1.5 Admission Requirements

Admission to Constructor University is selective and based on a candidate’s school and/or university achievements, recommendations, self-presentation, and performance on standardized tests. Students admitted to Constructor University demonstrate exceptional academic achievements, intellectual creativity, and the desire and motivation to make a difference in the world.

The following documents need to be submitted with the application:

- Recommendation Letter (optional)
- Official or certified copies of high school/university transcripts
- Educational History Form
- Standardized test results (SAT/ACT) if applicable
- Motivation statement
- ZeeMee electronic resume (optional)
- Language proficiency test results (TOEFL Score: 90, IELTS: Level 6.5 or equivalent)

Formal admission requirements are subject to higher education law and are outlined in the Admission and Enrollment Policy of Constructor University.

For more detailed information about the admission visit:
https://constructor.university/admission-aid/application-information-undergraduate

1.6 More Information and Contacts

For more information on the study program, please contact the Study Program Coordinator:

Prof. Dr. Andreas Birk

Professor of Electrical Engineering & Computer Science

Email: abirk@constructor.university

or visit our program website: https://constructor.university/programs/online-programs/applied-computer-science

For more information on Student Services, please visit:

https://constructor.university/student-life/student-services

2 The Curricular Structure

2.1 General

The curricular structure provides multiple elements for enhancing employability, interdisciplinarity, and internationality. Additionally, a mandatory internship (or work in a start-up) of at least two months after the second year of study gives students opportunities to gain insight into the professional world, apply their intercultural competences and reflect on their roles and ambitions for employment and in a globalized society.

All undergraduate programs at Constructor University are based on a coherently modularized structure, which provides students with a certain degree of flexibility regarding their individual study path and which ensures that they can complete their studies within the regular period.

The framework policies and procedures regulating undergraduate study programs at Constructor University can be found on the website (https://constructor.university/student-life/student-services/university-policies/academic-policies).

2.2 The Curriculum

2.2.1 Year 1

The first study year is characterized by a university-specific offering of disciplinary education that builds on and expands upon the students' entrance qualifications. Students take introductory modules for a total of 60 CP from the Year 1 area. The team of Academic Advising Services offers curriculum counseling to all Bachelor students independently of their major, while Academic Advisors, in their capacity as contact persons from the faculty, support students individually in deciding on their major study program.

Applied Computer Science students take the following mandatory (m) modules in the first semester (30 CP)

- CHOICE Module: Introduction to Computer Science (m, 7.5 CP)
- CHOICE Module: Programming in C/C++ (m, 7.5 CP)
- CHOICE Module: Introduction to Data Science (m, 7.5 CP)
- CHOICE Module: Distributed Development (m, 2.5 CP)
- Methods Module: Elements of Linear Algebra (m, 5 CP)

and the following modules in the second semester (30 CP):

- CHOICE Module: Algorithms and Data Structures (m, 7.5 CP)
- CHOICE Module: Introduction to Cyber Physical Systems (m, 7.5 CP)
- CHOICE Module: Software Design and Prototyping (m, 7.5 CP)
- CHOICE Module: Distributed Development (m, 2.5 CP)
- Methods Module: Elements of Calculus (m, 5 CP)

The modules Programming in C and C++ and Algorithms and Data Structures introduce students to imperative and object-oriented programming and basic algorithms and data structures. The Introduction to Computer Science module discusses abstract and concrete notions of computing machines and algorithms, and the representation of information. Students are also exposed to a pure functional programming language. The Software Design and Prototyping module deals with prototyping software, also known as mockup systems. It is complemented by the Distributed Development module that deals with practical aspects of remotely developing software in teams distributed at different physical locations. The module Introduction to Cyber Physical Systems deals with the relations and interfaces of software to computer hardware, embedded systems, sensors and actuators, and networking. Relevant mathematical content is covered in the Matrix Algebra and Advanced Calculus modules and in the Introduction to Data Science module.

2.2.2 Year 2

In their second year, students take a total of 50 CP from a selection of in-depth, discipline-specific modules. Building on the introductory Year 1 modules and applying the methods and skills students have already acquired so far, these modules aim to expand the students' critical understanding of the key theories, principles, and methods in their major for the current state of knowledge and best practice.

In Year 2, Applied Computer Science students acquire the following disciplinary and methods mandatory modules (50 CP in total):

- CORE Module: Databases and Web Services (m, 7.5 CP)
- CORE Module: Operating Systems (m, 7.5 CP)
- CORE Module: Data Analytics and Modeling (m, 7.5 CP)
- CORE Module: Software Engineering (m, 7.5 CP)
- CORE Module: Artificial Intelligence (m, 7.5 CP)
- CORE Module: Machine Learning (m, 7.5 CP)
- Methods Module: Probability and Random Processes (m, 5 CP)

In the second year, core areas of Computer Science with a high relevance to modern software development are covered in the modules Databases and Web Services, Operating Systems, Artificial Intelligence, and Machine Learning. Knowledge in Software Engineering is deepened in the according module. Relevant mathematical aspects are covered in the modules Probability and Random Processes and Data Analytics, where the latter – together with Artificial Intelligence and Machine Learning – also deepens the knowledge related to Data Science. Multiple modules include practical software development aspects, namely Software Engineering, Databases and Web Services, Artificial Intelligence, Machine Learning, Machine Learning Tools and Data Analytics and Modeling.

Additionally, the students will take two mandatory "new skills" modules from the university-wide CONSTRUCTOR Track which is dedicated to multidisciplinary content dedicated to methods as well as intellectual skills (5 CP in total) (please also see 2.2.3.4):

- New Skills Module: Logic (m, 2.5 CP)
- New Skills Module: Causation and Correlation (m, 2.5 CP)

The remaining 5 CP must be chosen from the Management Elective area (mandatory elective, me), which includes Management oriented modules that provide basic business and management knowledge.

- Management Module: Digital Business Models and Functions (me, 5 CP)
- Management Module: Digital Marketing Fundamentals and SEO (me, 5 CP)

An updated list of all modules in this Elective area will be available in the online course catalogue at the start of the second academic year.

2.2.3 Year 3

During their third year, students prepare for and make decisions about their career after graduation. To explore available choices fitting individual interests, and to gain professional experience, students take a mandatory summer internship (see 2.2.3.1). The third year of studies allows Applied Computer Science students to take ACS Specialization modules, two new skills modules and two further Management Elective modules (as described in Chapter 2.2.3.3). Finally, the 6th semester is dedicated to fostering the students' research experience by involving them in a Bachelor thesis project.

2.2.3.1 Internship/Startup and Career Skills Module

As a core element of Constructor University's employability approach students are required to engage in a mandatory two-month internship of 15 CP that will usually be completed during the summer between the second and third years of study. This gives students the opportunity to gain first-hand practical experience in a professional environment, apply their knowledge and understanding in a professional context, reflect on the relevance of their major to employment and society, reflect on their own personal role in employment and society, and develop a professional orientation. The internship can also establish valuable contacts for the students' bachelor's thesis project, for the selection of a master program graduate school or further employment after graduation. This module is complemented by career advising and several career skills workshops throughout all six semesters that prepare students for the transition from student life to professional life. As an alternative to the full-time internship, students interested in setting up their own company can apply for a start-up option to focus on developing their business plans.

For further information, please contact the Career Service Center (CSC)
(https://constructor.university/student-life/career-services).

2.2.3.2 ACS Specialization Modules

In the third year of their studies, students take 15 CP of advanced ACS Specialization modules to consolidate their knowledge and to be exposed to state-of-the-art research in the areas of their interest. This curricular component is offered as a portfolio of modules, from among which students can select freely during their fifth and sixth semester. The default module size is 5 CP, with smaller 2.5 CP modules being possible as justified exceptions.

Applied Computer Science students take at least 15 CP from the following abridged list of ACS Specialization Modules:

- ACS Specialization Module: Computer Graphics (me, 5 CP)
- ACS Specialization Module: Computer Networks (me, 5 CP)

- ACS Specialization Module: Web Application Development (me, 5 CP)
- ACS Specialization Module: Human Computer Interaction (me, 5 CP)

An updated list of all modules in the ACS Specialization area will be available in the online course catalogue at the start of the third academic year.

2.2.3.3 Management Modules

Students take 5 CP from the Management area to acquire valuable knowledge in the field of business and management. Modules in this area aim to bridge the gap from software development to marketable software products and to prepare students interested in a management-oriented career track. A broad spectrum of topics is tackled, such as product development, innovation, marketing, leadership, general business, and change management. An updated list of all modules in the Management area will be available in the online course catalogue at the start of the third academic year.

2.2.3.4 Collaborative Software Project

In the collaborative software project (m, 5 CP), the students deepen their knowledge and skills in one or multiple areas of the first and especially second year. They are exposed to state-of-the-art research with the goal to derive ideas and strategies to address application-oriented problems and to develop software for them. Students learn how to organize and execute an application-oriented research and development (R&D) project. Students are expected to organize themselves in group work under the guidance of the instructor.

2.2.3.5 New Skills

This part of the curriculum constitutes the intellectual and conceptual tool kit, and is designed to cultivate and nurture the capacity for a particular set of intellectual dispositions – curiosity, imagination, critical thought, transferability – as well as a range of individual and societal capacities – self-reflection, argumentation and communication – and to introduce students to the normative aspects of inquiry and research – including the norms governing sourcing, sharing, withholding materials and research results as well as others governing the responsibilities of expertise as well as the professional point of view.

All students are required to take the following modules in their second year as mentioned in 2.2.2:

- New Skills Module: Logic (m, 2.5 CP)
- New Skills Module: Causation and Correlation (m, 2.5 CP)

In the third year, students take two 5 CP modules that build upon previous modules in the track and are partially constituted by modules that are more closely linked to each student's disciplinary field of study. The following modules are mandatory for all students in the third year:

- New Skills Module: Argumentation, Data Visualization and Communication (m, 5 CP)
- New Skills Module: Agency, Leadership and Accountability (m, 5 CP)

2.2.3.6 Bachelor Thesis

This module is a mandatory graduation requirement for all undergraduate students. The title of the thesis will appear on the students' transcripts.

Within this module, students apply the knowledge skills, and methods they have acquired in their major discipline to become acquainted with actual research topics, ranging from the identification of suitable (short-term) research projects, preparatory literature searches, the realization of discipline-specific research, and the documentation, discussion, and interpretation of the results.

With their Bachelor Thesis students demonstrate mastery of the contents and methods of their major-specific research field. Furthermore, students show the ability to analyze and solve a well-defined problem with scientific approaches, a critical reflection of the status quo in scientific literature, and the original development of their own ideas. With the permission of a Constructor Faculty Supervisor, the Bachelor Thesis can also have an interdisciplinary nature.

3 Applied Computer Science Undergraduate Program Regulations

3.1 Scope of these Regulations

The regulations in this handbook are valid for all students who entered the Applied Computer Science undergraduate program at Constructor University in Fall 2025. In case of a conflict between the regulations in this handbook and the general Policies for Bachelor Studies, the latter apply (see (https://constructor.university/student-life/student-services/university-policies/academic-policies).

In exceptional cases, certain necessary deviations from the regulations of this study handbook might occur during the course of study (e.g., change of the semester sequence, assessment type, or the teaching mode of courses).

Updates to Study Program Handbooks are based on the policies approved by the Academic Senate on substantial and nonsubstantial changes to study programs. Students are integrated in the decision-making process through their respective committee representatives. All students affected by the changes will be properly informed.

In general, Constructor University therefore reserves the right to change or modify the regulations of the program handbook also after its publication at any time and in its sole discretion.

3.2 Degree

Upon successful completion of this study program, students are awarded a Bachelor of Science degree in Applied Computer Science.

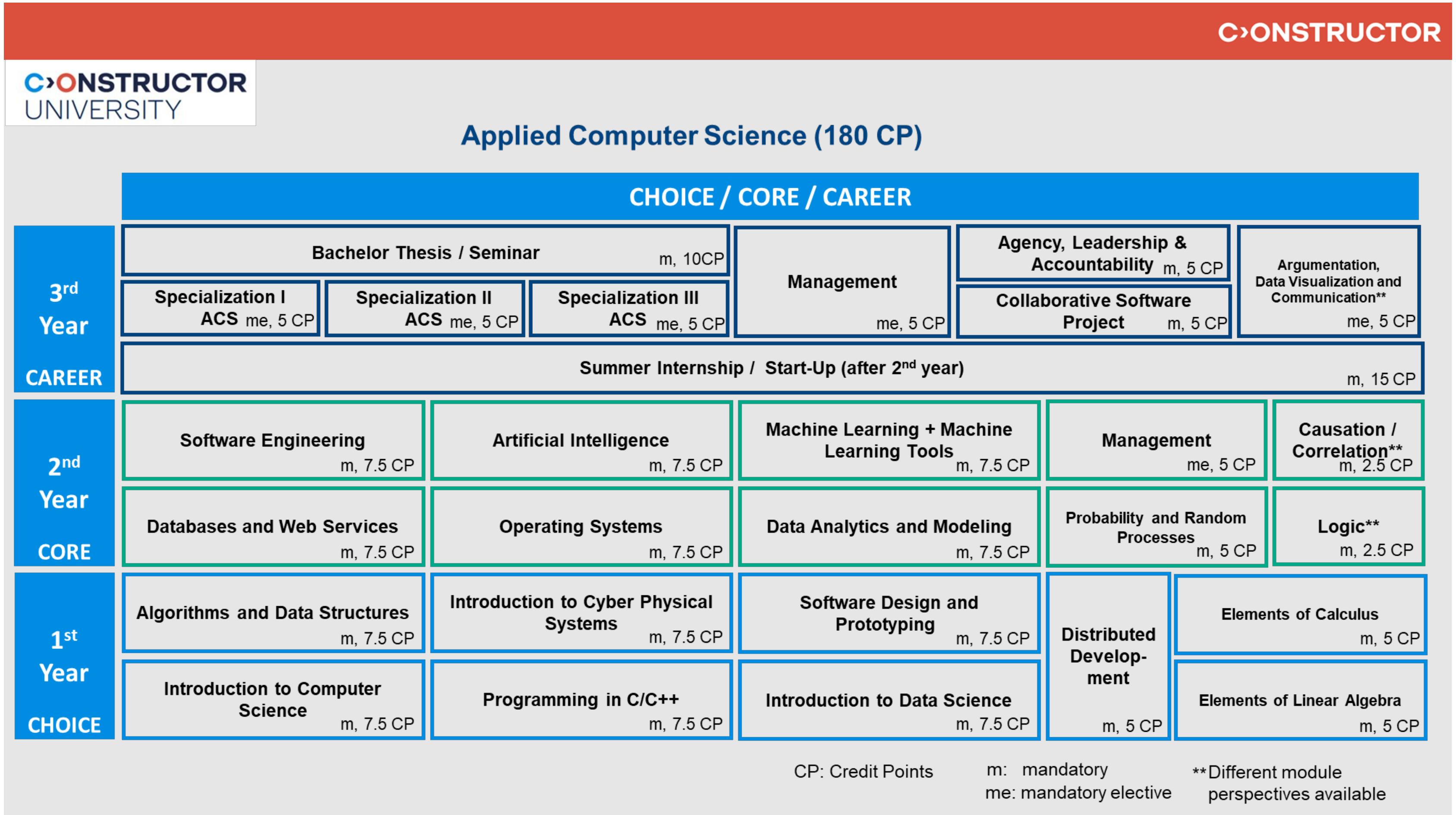
3.3 Graduation Requirements

In order to graduate, students need to obtain 180 CP. In addition, the following graduation requirements apply:

Students need to complete all mandatory components of the program as indicated in the Study and Examination Plan in Chapter 5 of this handbook.

4 Schematic Study Plan for Applied Computer Science

Figure 1 shows schematically the sequence and types of modules required for the study program. A more detailed description, including the assessment types, is given in the Study and Examination Plans in the following section.



C›ONSTRUCTOR

C›ONSTRUCTOR UNIVERSITY

Applied Computer Science (180 CP)

CHOICE / CORE / CAREER

3rd Year CAREER

- Bachelor Thesis / Seminar m, 10CP
- Specialization I ACS me, 5 CP
- Specialization II ACS me, 5 CP
- Specialization III ACS me, 5 CP
- Management me, 5 CP
- Agency, Leadership & Accountability m, 5 CP
- Collaborative Software Project m, 5 CP
- Argumentation, Data Visualization and Communication** me, 5 CP
- Summer Internship / Start-Up (after 2nd year) m, 15 CP

2nd Year CORE

- Software Engineering m, 7.5 CP
- Artificial Intelligence m, 7.5 CP
- Machine Learning + Machine Learning Tools m, 7.5 CP
- Management me, 5 CP
- Causation / Correlation** m, 2.5 CP
- Databases and Web Services m, 7.5 CP
- Operating Systems m, 7.5 CP
- Data Analytics and Modeling m, 7.5 CP
- Probability and Random Processes m, 5 CP
- Logic** m, 2.5 CP

1st Year CHOICE

- Algorithms and Data Structures m, 7.5 CP
- Introduction to Cyber Physical Systems m, 7.5 CP
- Software Design and Prototyping m, 7.5 CP
- Distributed Develop-ment m, 5 CP
- Elements of Calculus m, 5 CP
- Introduction to Computer Science m, 7.5 CP
- Programming in C/C++ m, 7.5 CP
- Introduction to Data Science m, 7.5 CP
- Elements of Linear Algebra m, 5 CP

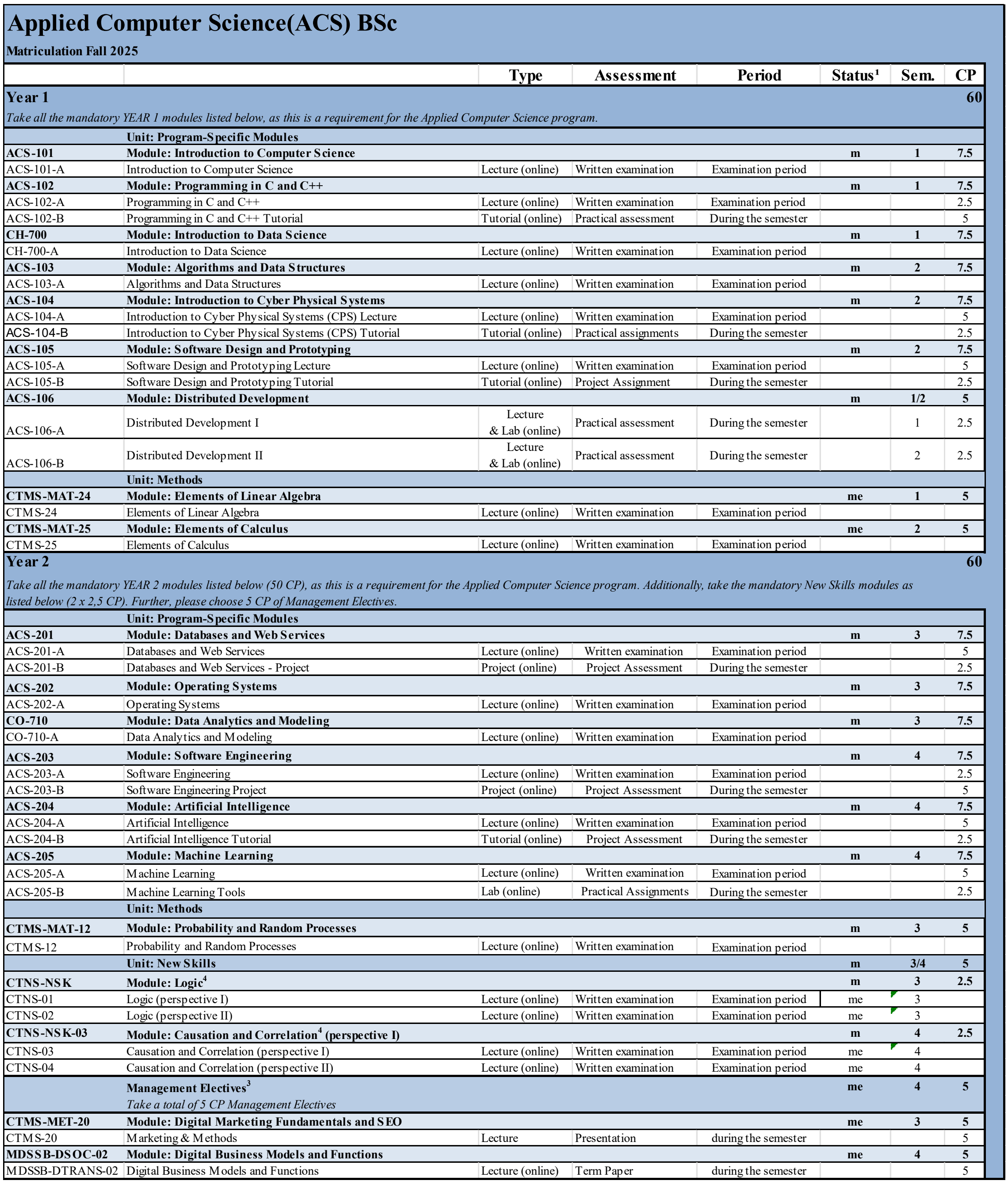
CP: Credit Points

m: mandatory
me: mandatory elective

**Different module perspectives available

Figure 1: Schematic Study Plan for Applied Computer Science

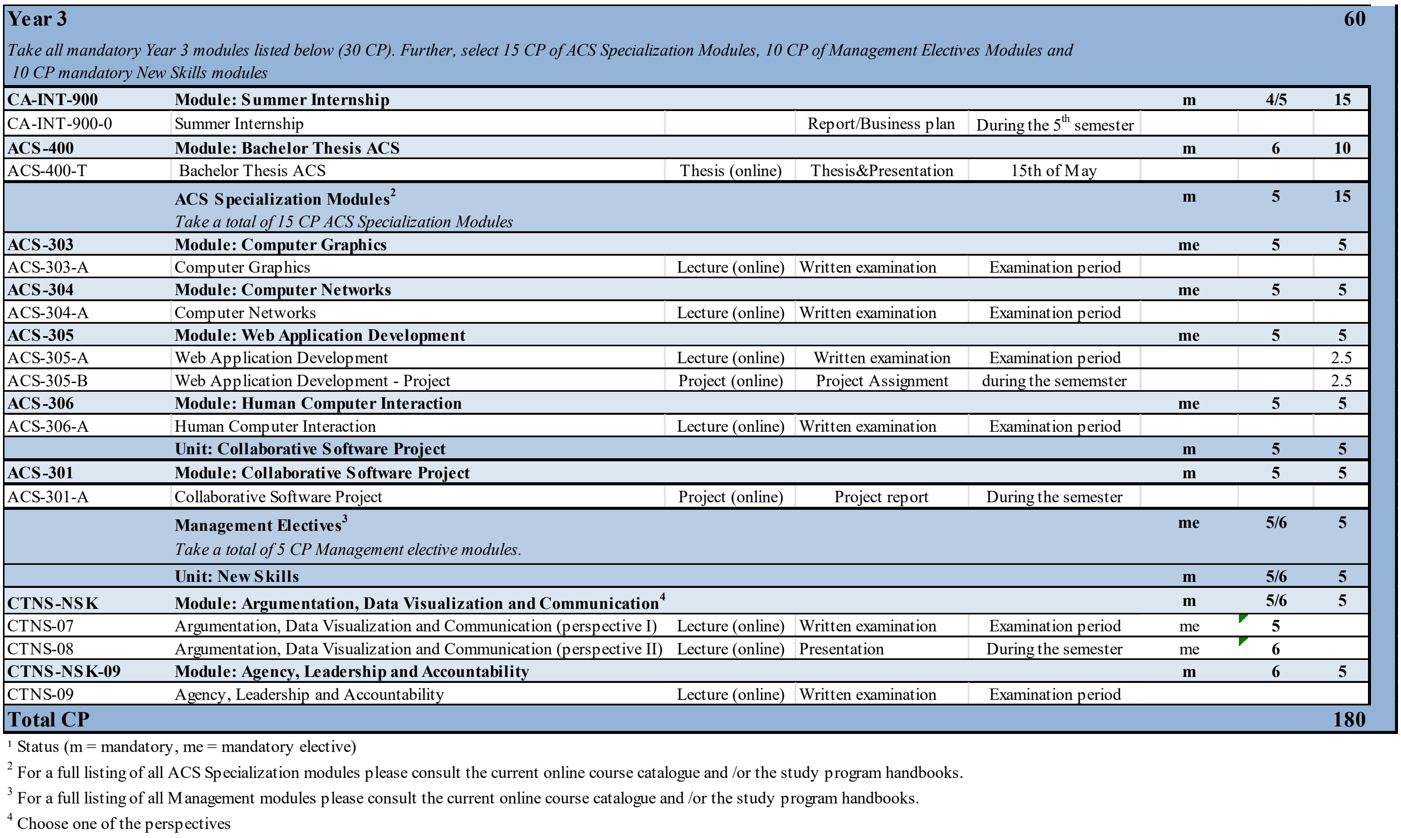
5 Study and Examination Plan



Applied Computer Science(ACS) BSc

Matriculation Fall 2025

		Type	Assessment	Period	Status[1]	Sem.	CP
Year 1							**60**
Take all the mandatory YEAR 1 modules listed below, as this is a requirement for the Applied Computer Science program.							
	Unit: Program-Specific Modules						
ACS-101	**Module: Introduction to Computer Science**				**m**	**1**	**7.5**
ACS-101-A	Introduction to Computer Science	Lecture (online)	Written examination	Examination period			
ACS-102	**Module: Programming in C and C++**				**m**	**1**	**7.5**
ACS-102-A	Programming in C and C++	Lecture (online)	Written examination	Examination period			2.5
ACS-102-B	Programming in C and C++ Tutorial	Tutorial (online)	Practical assessment	During the semester			5
CH-700	**Module: Introduction to Data Science**				**m**	**1**	**7.5**
CH-700-A	Introduction to Data Science	Lecture (online)	Written examination	Examination period			
ACS-103	**Module: Algorithms and Data Structures**				**m**	**2**	**7.5**
ACS-103-A	Algorithms and Data Structures	Lecture (online)	Written examination	Examination period			
ACS-104	**Module: Introduction to Cyber Physical Systems**				**m**	**2**	**7.5**
ACS-104-A	Introduction to Cyber Physical Systems (CPS) Lecture	Lecture (online)	Written examination	Examination period			5
ACS-104-B	Introduction to Cyber Physical Systems (CPS) Tutorial	Tutorial (online)	Practical assignments	During the semester			2.5
ACS-105	**Module: Software Design and Prototyping**				**m**	**2**	**7.5**
ACS-105-A	Software Design and Prototyping Lecture	Lecture (online)	Written examination	Examination period			5
ACS-105-B	Software Design and Prototyping Tutorial	Tutorial (online)	Project Assignment	During the semester			2.5
ACS-106	**Module: Distributed Development**				**m**	**1/2**	**5**
ACS-106-A	Distributed Development I	Lecture & Lab (online)	Practical assessment	During the semester		1	2.5
ACS-106-B	Distributed Development II	Lecture & Lab (online)	Practical assessment	During the semester		2	2.5
	Unit: Methods						
CTMS-MAT-24	**Module: Elements of Linear Algebra**				**me**	**1**	**5**
CTMS-24	Elements of Linear Algebra	Lecture (online)	Written examination	Examination period			
CTMS-MAT-25	**Module: Elements of Calculus**				**me**	**2**	**5**
CTMS-25	Elements of Calculus	Lecture (online)	Written examination	Examination period			
Year 2							**60**
Take all the mandatory YEAR 2 modules listed below (50 CP), as this is a requirement for the Applied Computer Science program. Additionally, take the mandatory New Skills modules as listed below (2 x 2,5 CP). Further, please choose 5 CP of Management Electives.							
	Unit: Program-Specific Modules						
ACS-201	**Module: Databases and Web Services**				**m**	**3**	**7.5**
ACS-201-A	Databases and Web Services	Lecture (online)	Written examination	Examination period			5
ACS-201-B	Databases and Web Services - Project	Project (online)	Project Assessment	During the semester			2.5
ACS-202	**Module: Operating Systems**				**m**	**3**	**7.5**
ACS-202-A	Operating Systems	Lecture (online)	Written examination	Examination period			
CO-710	**Module: Data Analytics and Modeling**				**m**	**3**	**7.5**
CO-710-A	Data Analytics and Modeling	Lecture (online)	Written examination	Examination period			
ACS-203	**Module: Software Engineering**				**m**	**4**	**7.5**
ACS-203-A	Software Engineering	Lecture (online)	Written examination	Examination period			2.5
ACS-203-B	Software Engineering Project	Project (online)	Project Assessment	During the semester			5
ACS-204	**Module: Artificial Intelligence**				**m**	**4**	**7.5**
ACS-204-A	Artificial Intelligence	Lecture (online)	Written examination	Examination period			5
ACS-204-B	Artificial Intelligence Tutorial	Tutorial (online)	Project Assessment	During the semester			2.5
ACS-205	**Module: Machine Learning**				**m**	**4**	**7.5**
ACS-205-A	Machine Learning	Lecture (online)	Written examination	Examination period			5
ACS-205-B	Machine Learning Tools	Lab (online)	Practical Assignments	During the semester			2.5
	Unit: Methods						
CTMS-MAT-12	**Module: Probability and Random Processes**				**m**	**3**	**5**
CTMS-12	Probability and Random Processes	Lecture (online)	Written examination	Examination period			
	Unit: New Skills				**m**	**3/4**	**5**
CTNS-NSK	**Module: Logic[4]**				**m**	**3**	**2.5**
CTNS-01	Logic (perspective I)	Lecture (online)	Written examination	Examination period	me	3	
CTNS-02	Logic (perspective II)	Lecture (online)	Written examination	Examination period	me	3	
CTNS-NSK-03	**Module: Causation and Correlation[4] (perspective I)**				**m**	**4**	**2.5**
CTNS-03	Causation and Correlation (perspective I)	Lecture (online)	Written examination	Examination period	me	4	
CTNS-04	Causation and Correlation (perspective II)	Lecture (online)	Written examination	Examination period	me	4	
	Management Electives[3] *Take a total of 5 CP Management Electives*				**me**	**4**	**5**
CTMS-MET-20	**Module: Digital Marketing Fundamentals and SEO**				**me**	**3**	**5**
CTMS-20	Marketing & Methods	Lecture	Presentation	during the semester			5
MDSSB-DSOC-02	**Module: Digital Business Models and Functions**				**me**	**4**	**5**
MDSSB-DTRANS-02	Digital Business Models and Functions	Lecture (online)	Term Paper	during the semester			5



Year 3							**60**
Take all mandatory Year 3 modules listed below (30 CP). Further, select 15 CP of ACS Specialization Modules, 10 CP of Management Electives Modules and 10 CP mandatory New Skills modules							
CA-INT-900	**Module: Summer Internship**				**m**	**4/5**	**15**
CA-INT-900-0	Summer Internship		Report/Business plan	During the 5th semester			
ACS-400	**Module: Bachelor Thesis ACS**				**m**	**6**	**10**
ACS-400-T	Bachelor Thesis ACS	Thesis (online)	Thesis&Presentation	15th of May			
	ACS Specialization Modules[2] *Take a total of 15 CP ACS Specialization Modules*				**m**	**5**	**15**
ACS-303	**Module: Computer Graphics**				**me**	**5**	**5**
ACS-303-A	Computer Graphics	Lecture (online)	Written examination	Examination period			
ACS-304	**Module: Computer Networks**				**me**	**5**	**5**
ACS-304-A	Computer Networks	Lecture (online)	Written examination	Examination period			
ACS-305	**Module: Web Application Development**				**me**	**5**	**5**
ACS-305-A	Web Application Development	Lecture (online)	Written examination	Examination period			2.5
ACS-305-B	Web Application Development - Project	Project (online)	Project Assignment	during the sememster			2.5
ACS-306	**Module: Human Computer Interaction**				**me**	**5**	**5**
ACS-306-A	Human Computer Interaction	Lecture (online)	Written examination	Examination period			
	Unit: Collaborative Software Project				**m**	**5**	**5**
ACS-301	**Module: Collaborative Software Project**				**m**	**5**	**5**
ACS-301-A	Collaborative Software Project	Project (online)	Project report	During the semester			
	Management Electives[3] *Take a total of 5 CP Management elective modules.*				**me**	**5/6**	**5**
	Unit: New Skills				**m**	**5/6**	**5**
CTNS-NSK	**Module: Argumentation, Data Visualization and Communication**[4]				**m**	**5/6**	**5**
CTNS-07	Argumentation, Data Visualization and Communication (perspective I)	Lecture (online)	Written examination	Examination period	me	5	
CTNS-08	Argumentation, Data Visualization and Communication (perspective II)	Lecture (online)	Presentation	During the semester	me	6	
CTNS-NSK-09	**Module: Agency, Leadership and Accountability**				**m**	**6**	**5**
CTNS-09	Agency, Leadership and Accountability	Lecture (online)	Written examination	Examination period			
Total CP							**180**

[1] Status (m = mandatory, me = mandatory elective)

[2] For a full listing of all ACS Specialization modules please consult the current online course catalogue and /or the study program handbooks.

[3] For a full listing of all Management modules please consult the current online course catalogue and /or the study program handbooks.

[4] Choose one of the perspectives

Figure 2: Study and Examination Plan

6 Modules

6.1 Introduction to Computer Science

Module Name	**Introduction to Computer Science**
Module Code	2025-ACS-101
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 1 - 2025-S-ACS-BSc 1 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Jürgen Schönwälder

Forms of Learning and Teaching	
Asynchronous Self Study	52.5
Interactive Learning	10
Exam Preparation	115
Independent Study	10
Workload Hours	187.5 hours

Module Components	**Number**	**Type**	**CP**
Introduction to Computer Science	ACS-101-A	Lecture (Online)	7.5

Module Description

The module introduces fundamental concepts and techniques of computer science in a bottom-up manner. Based on clear mathematical foundations (which are developed as needed), the course discusses abstract and concrete notions of computing machines, information, and algorithms, focusing on the question of representation versus meaning in Computer Science.

The module introduces basic concepts of discrete mathematics with a focus on inductively defined structures, to develop a theoretical notion of computation. Students will learn the basics of the functional programming language Haskell because it treats computation as the evaluation of pure and typically inductively defined functions. The module covers a basic subset of Haskell that includes types, recursion, tuples, lists, strings, higher-order functions, and finally monads. Back on the theoretical side, the module covers the syntax and semantics of Boolean expressions and it explains how Boolean algebra relates to logic gates and digital circuits. On the technical side, the course introduces the representation of basic data types such as numbers, characters, and strings as well as the von Neuman computer architecture. On the algorithmic side, the course introduces the notion of correctness and elementary concepts of complexity theory (big O notation).

Recommended Knowledge

It is recommended that students install a Linux system such as Ubuntu on their notebooks and that they become familiar with basic tools such as editors (vim or emacs) and the basics of a shell. The Glasgow Haskell Compiler (GHC) will be used for implementing Haskell programs.

Usability and Relationship to other Modules

This module introduces key mathematical concepts and various notions of computing machines and computing abstractions and is in particularly important for subsequent courses covering theoretical aspects of computer science. This module is also important for courses that require a basic understanding of computer architecture and program execution at the hardware level.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain basic concepts such as the correctness and complexity of algorithms (including the big O notation)
2	Illustrate	Illustrate basic concepts of discrete math (sets, relations, functions)
3	Recall	Recall basic proof techniques and use them to prove properties of algorithms
4	Explain	Explain the representation of numbers (integers, floats), characters and strings, and date and time
5	Summarize	Summarize basic principles of Boolean algebra and Boolean logic
6	Describe	Describe how Boolean logic relates to logic gates and digital circuits
7	Outline	Outline the basic structure of a von Neumann computer
8	Explain	Explain the execution of machine instructions on a von Neumann computer
9	Describe	Describe the difference between assembler languages and higher-level programming languages
10	Define	Define the differences between interpretation and compilation
11	Illustrate	Illustrate how an operating system kernel supports the execution of programs
12	Determine	Determine the correctness of simple programs
13	Write	Write simple programs in a pure functional programming language

Indicative Literature

- Eric Lehmann, F. Thomson Leighton, Albert R. Meyer: Mathematics for Computer Science, online 2018.
- David A. Patterson, John L Hennessy: Computer Organization and Design: The Hardware/Software Interface, 6th edition, Morgan Kaufmann, 2020
- Miran Lipovaca: Learn You a Haskell for Great Good!: A Beginner's Guide, 1st edition, No Starch Press, 2011.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Introduction to Computer Science	Written Examination	120 minutes	100	45%	All intended learning outcomes of the module

Module Achievements: 50% of the assignments correctly solved

6.2 Programming in C and C++

Module Name	Programming in C and C++
Module Code	2025-ACS-102
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 1 - 2025-S-ACS-BSc 1 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Dr. Kinga Lipskoch

Forms of Learning and Teaching	
Asynchronous Self Study	17.5
Interactive Learning	92.5
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Programming in C and C++	ACS-102-A	Lecture (Online)	2.5
Programming in C and C++ Tutorial	ACS-102-B	Tutorial (Online)	5

Module Description

This course offers an introduction to programming using the programming languages C and C++. After a short overview of the program development cycle (editing, preprocessing, compiling, linking, executing), the module presents the basics of C programming. Fundamental imperative programming concepts such as variables, loops, and function calls are introduced in a hands-on manner. Afterwards, basic data structures such as multidimensional arrays, structures, and pointers are introduced and dynamically allocated multidimensional arrays and linked lists and trees are used for solving simple practical problems. The relationships between pointers and arrays, pointers and structures, and pointers and functions are described, and they are illustrated using examples that also introduce recursive functions, file handling, and dynamic memory allocation.

The module then introduces basic concepts of object-oriented programming languages using the programming language C++ in a hands-on manner. Concepts such as classes and objects, data abstractions, and information hiding are introduced. C++ mechanisms for defining and using objects, methods, and operators are introduced and the relevance of constructors, copy constructors, and destructors for dynamically created objects is explained. Finally, concepts such as inheritance, polymorphism, virtual functions, and overloading are introduced. The learned concepts are applied by solving programming problems.

Recommended Knowledge

It is recommended that students install a suitable programming environment on their notebooks. It is recommended to install a Linux system such as Ubuntu, which comes with open-source compilers such as gcc and g++ and editors such as vim or emacs. Alternatively, the open-source Code: Blocks integrated development environment can be installed to solve programming problems.

Usability and Relationship to other Modules

This module introduces the programming languages C and C++ and several other modules build on this foundation. Certain features of C++ such as templates and generic data structures and an overview of the standard template library will be covered in the Algorithms and Data Structures module.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain basic concepts of imperative programming languages such as variables, assignments, loops, and function calls
2	Write	Write, test, and debug programs in the procedural programming language C using basic C library functions
3	Demonstrate	Demonstrate how to use pointers to create dynamically allocated data structures such as linked lists
4	Explain	Explain the relationship between pointers and arrays
5	Illustrate	Illustrate basic object-oriented programming concepts such as objects, classes, information hiding, and inheritance
6	Give	Give original examples of function and operator overloading and polymorphism
7	Write	Write, test, and debug programs in the object-oriented programming language C++

Indicative Literature

- Brian Kernighan, Dennis Ritchie: The C Programming Language, 2nd edition, Prentice Hall Professional Technical Reference, 1988.
- Steve Oualline: Practical C Programming, 3rd edition, O'Reilly Media, 1997.
- Bruce Eckel: Thinking in C++: Introduction to Standard C++, Prentice Hall, 2000.
- Bruce Eckel, Chuck Allison: Thinking in C++: Practical Programming, Prentice Hall, 2004.
- Bjarne Stroustrup: The C++ Programming Language, 4th edition, Addison Wesley, 2013.
- Michael Dawson: Beginning C++ Through Game Programming, 4th edition, Delmar Learning, 2014.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs

Programming in C and C++	Written Examination	120 minutes	33	45%	All theoretical ILOs of the module
Programming in C and C++ Tutorial	Practical Assessment		67	45%	All theoretical ILOs of the module

Module Achievements: None

6.3 Introduction to Data Science

Module Name	Introduction to Data Science
Module Code	2025-CH-700
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 1 - 2025-S-ACS-BSc 2 - 2025-Minor-Data-Science 1 - 2025-MDDA-BSc 1 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-Minor-Data-Science (Minor in Data Science)
Module Coordinator(s)	Prof. Dr. Hilke Brockmann

Forms of Learning and Teaching	
Asynchronous Self Study	52.5
Interactive Learning	57.5
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Introduction to Data Science	CH-700-A	Lecture (Online)	7.5

Module Description

The module introduces data science with an integrated presentation of three essential components, namely, (1) societal/legal implications and business opportunities, (2) technical/theoretical background and case studies, (3) an introduction to the Python coding environment. The first component entails a conceptual introduction to the opportunities and the challenges of a digitally transformed and data-driven society, presentations on industry standards and legal frameworks, and discussions of critical issues such as cybersecurity and surveillance. The second component includes topics such as data science terminology, digital data and their representations, and introductions to exploratory data analysis and prominent supervised and unsupervised learning tasks. The third component offers an introduction to the Python ecosystem of data representation, processing, analysis, and visualization, starting with Jupyter notebooks, installing suitable environments, and introductions to data science related packages such as NumPy, SciPy, Matplotlib, Seaborn, and Pandas. Fundamental data science concepts are summarized and illustrated using real-world data from various disciplines. Flexible educational formats (mostly online and hybrid) allow for asynchronous learning. Lectures are combined with an exposure to Python programming and data processing and visualization environments, including hands-on practicals, examples, and exercises.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain societal implications of the digital transformation.
2	Understand	Understand the legal data protection framework.
3	Carry	Carry out basic data processing and visualization tasks.
4	Apply	Apply fundamental data science methods to structured data.
5	Understand	Understand the logic of Python scripts and functions.
6	Compose	Compose Python code using templates.

Indicative Literature

- Ani Adhikari, John DeNero, David Wagner. Computational and Inferential Thinking: The Foundations of Data Science. Originally developed for the UC Berkeley course Data 8: Foundations of Data Science. An online version of the textbook is available at https://inferentialthinking.com/.
- The Alan Turing Institute, Data Science for the Social Good.
- Philip D. Brooker. Programing with Python for Social Scientists. Sage 2020.
- Shin Takahasi, Iroha Inoue. The Manga Guide to Linear Algebra. Trend-Pro 2012.
- Steven S. Skiena. The Data Science Design Manual. Springer 2017.
- Jake Vanderplas. Python Data Science Handbook. O'Reilly 2016. An online version is available at https://jakevdp.github.io/PythonDataScienceHandbook/.
- Shoshana Zuboff. The Age of Surveillance Capitalism. London: Profile 2019.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Introduction to Data Science	Written Examination	180 minutes	100	45%	1-6

Module Achievements: 50% of the assignments need to be correctly solved.

6.4 Algorithms and Data Structures

Module Name	Algorithms and Data Structures
Module Code	2025-ACS-103
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 2 - 2025-S-ACS-BSc 2 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Dr. Kinga Lipskoch

Forms of Learning and Teaching	
Asynchronous Self Study	52.5
Interactive Learning	57.5
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Algorithms and Data Structures	ACS-103-A	Lecture (Online)	7.5

Module Description

Algorithms and data structures are the core of computer science. An algorithm is an effective description for calculations using a finite list of instructions that can be executed by a computer. A data structure is a concept for organizing data in a computer such that data can be used efficiently. This introductory module allows students to learn about fundamental algorithms for solving problems efficiently. It introduces basic algorithmic concepts; fundamental data structures for efficiently storing, accessing, and modifying data; and techniques that can be used for the analysis of algorithms and data structures with respect to their computational and memory complexities. The presented concepts and techniques form the basis of almost all computer programs.

Recommended Knowledge

Students should refresh their knowledge of the C and C++ programming language and be able to solve simple programming problems in C and C++. Students are expected to have a working programming environment.

Usability and Relationship to other Modules

Familiarity with basic algorithms and data structures is fundamental for almost all advanced modules in computer science. This module additionally introduces advanced concepts of the C++ programming language that are needed in advanced programming-oriented modules in the 2nd and 3rd years of the CS and RIS programs.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain asymptotic (time and memory) complexities and respective notations
2	Able	Able to prove asymptotic complexities of algorithms
3	Illustrate	Illustrate basic data structures such as arrays, lists, queues, stacks, trees, and hash tables
4	Describe	Describe algorithmic design concepts and apply them to new problems
5	Explain	Explain basic algorithms (sorting, searching, graph algorithms, computational geometry) and their complexities
6	Summarize	Summarize and apply C++ templates and generic data structures provided by the standard C++ template library

Indicative Literature

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Introduction to Algorithms, 3rd edition, MIT Press, 2009.
- Donald E. Knuth: The Art of Computer Programming: Fundamental Algorithms, volume 1, 3rd edition, Addison Wesley Longman Publishing, 1997.

Entry Requirements

Prerequisites	Programming in C and C++
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Algorithms and Data Structures	Written Examination	120 minutes	100	45%	All intended learning outcomes of the module.

Module Achievements: None

6.5 Introduction to Cyber Physical Systems (CPS)

Module Name	Introduction to Cyber Physical Systems (CPS)
Module Code	2025-ACS-104
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 2 - 2025-S-ACS-BSc 1 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Prof. Dr. Andreas Birk

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	75
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Introduction to Cyber Physical Systems (CPS) Lecture	ACS-104-A	Lecture (Online)	5
Introduction to Cyber Physical Systems (CPS) Tutorial	ACS-104-B	Tutorial (Online)	2.5

Module Description

The area of Cyber Physical Systems (CPS) deals with the interface between the digital and the physical world, i.e., the relations and interfaces of software to computer hardware, embedded systems, sensors and actuators, and networking. Application examples range from large entities like power-grids, factories, or warehouses, down to smaller systems like automobiles, home automation, or machinery in production or warehouses. CPS builds on interconnected smart devices and intelligent autonomous systems, which may range from small simple sensor-nodes to more capable systems that may also features mobility and manipulation. It hence relates software development to aspects of computer architecture, communications, system integration, modelling, control, and artificial intelligence.

Recommended Knowledge

Students are expected to be familiar with the core elements of calculus and linear algebra.

Intended Learning Outcomes

No	Competence	ILO
1	Describe	Describe the different use-cases and application areas of CPS
2	Explain	Explain the components of CPS and their interplay

3	Understand	Understand computer architecture and be able to apply core concepts within embedded computing
4	Generate	Generate software interfaces to sensors and actuators
5	Understand	Understand the networking aspects related to CPS and apply them within the context of embedded computing
6	Explain	Explain real-time requirements and understand the related core software concepts and algorithms
7	Be	Be able to model systems
8	Understand	Understand and apply the basics of control of physical systems in form of software
9	Explain	Explain core concepts and methods of software for intelligent autonomous systems
10	Understand	Understand and use software methods for remote access for monitoring, operation, and maintenance of physical systems and processes

Indicative Literature

-

Entry Requirements

Prerequisites	Elements of Linear Algebra
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Introduction to Cyber Physical Systems (CPS) Lecture	Written Examination	120 minutes	67	45%	All intended learning outcomes of the module (with focus on theory).
Introduction to Cyber Physical Systems (CPS) Tutorial	Practical Assignments		33	45%	All intended learning outcomes of the module (with focus on theory).

Module Achievements: None

6.6 Software Design and Prototyping

Module Name	Software Design and Prototyping
Module Code	2025-ACS-105
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 2 - 2025-S-ACS-BSc 3 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Prof. Dr. Andreas Birk

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	75
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Software Design and Prototyping Lecture	ACS-105-A	Lecture (Online)	5
Software Design and Prototyping Tutorial	ACS-105-B	Tutorial (Online)	2.5

Module Description

During the early phases of software projects, it is often unclear what the exact requirements are and how a suitable software design could look like. Since wrong decisions taken during the early phases of a software project frequently have significant impact on the completion time and the overall costs of a software project, it is often desirable to quickly construct prototype systems. Prototype systems can not only be used to collect early feedback in order to clarify requirements. They can also be used to acquire additional customers. This module introduces software design pattern with a specific focus on the construction of early prototypes, sometimes also called mockup systems.

Recommended Knowledge

Students are expected to be familiar with programming in C/C++ and the basics of collaborative, remote software development.

Intended Learning Outcomes

No	Competence	ILO
1	Select	Select software architectures supporting fast prototyping
2	Implement	Implement interaction prototypes using suitable mockup tools
3	Implement	Implement backend and server prototypes using suitable mockup tools
4	Derive	Derive designs of interaction prototypes from incomplete user input
5	Conduct	Conduct an evaluation of mockup prototypes with target users

6	Be	Be able to revise prototypes efficiently in an agile manner
7	Effectively	Effectively work in a team prototyping different software components
8	Create	Create mock objects that can be used effectively for unit tests

Indicative Literature

-

Entry Requirements

Prerequisites	Programming in C and C++
Co-requisites	Distributed Development II
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Software Design and Prototyping Lecture	Written Examination	60 minutes	67	45%	Intended Learning outcomes 1,4 and 6.
Software Design and Prototyping Tutorial	Project Assessment		33	45%	Intended Learning outcomes 2,3,5,7 and 8

Module Achievements: None

6.7 Distributed Development

Module Name	**Distributed Development**
Module Code	2025-ACS-106
Module ECTS	5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 1 - 2025-F-ACS-BSc 2 - 2025-S-ACS-BSc 1 - 2025-S-ACS-BSc 2 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Prof. Dr. Andreas Birk

Forms of Learning and Teaching	
Asynchronous Self Study	17.5
Interactive Learning	52.5
Exam Preparation	20
Independent Study	35
Workload Hours	125 hours

Module Components	**Number**	**Type**	**CP**
Distributed Development I	ACS-106-A	Lecture and Laboratory (Online)	2.5
Distributed Development II	ACS-106-B	Lecture and Laboratory (Online)	2.5

Module Description

Software development is increasingly done in collaborative teams who work in a remote fashion, i.e., with team members who are spatially distributed at different locations, sometimes even across different time-zones. This can be very convenient for employers, who can recruit from around the globe without the need for expecting the employees to relocate, as well as for the employees, who gain some freedom in where and when they execute their tasks. But it includes also quite some challenges, e.g., for the development of a joined approach, the coordination of tasks, or the meeting of deadlines. This module provides a hands-on introduction into the methods and tools for handling these opportunities and challenges.

Recommended Knowledge

Previous experience with programming is a plus but not required.

Intended Learning Outcomes

No	**Competence**	**ILO**

1	Understand	Understand the opportunities and challenges that are involved in collaborative, remote software development
2	Comprehend	Comprehend the needs for and limitations of synchronous online-meeting tools
3	Use	Use the different standard features of tools for synchronous online-meetings
4	Comprehend	Comprehend the concepts of versioning software and be able to apply them
5	Understand	Understand the pro's and con's of asynchronous online communication
6	Use	Use standard features of online communication teams for brain-storming and the development of a joined approach to solve problems and the distribution of tasks
7	Understand	Understand the needs for calendars and to-do lists and how to handle them
8	Comprehend	Comprehend bug-trackers and be able to use them

Indicative Literature

-

Entry Requirements

Prerequisites	None
Co-requisites	Programming in C and C++
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Distributed Development I	Practical Assessment		50	45%	All intended learning outcomes of the module
Distributed Development II	Practical Assessment		50	45%	All intended learning outcomes of the module.

Module Achievements: None

6.8 Databases and Web Services

Module Name	Databases and Web Services
Module Code	2025-ACS-201
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 3 - 2025-S-ACS-BSc 3 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Peter Baumann

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	115
Exam Preparation	20
Independent Study	17.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Databases and Web Services	ACS-201-A	Lecture (Online)	5
Databases and Web Services - Project	ACS-201-B	Project (Online)	2.5

Module Description

This module offers a combined introduction to databases and web services. The database part starts with database design using the Entity Relationship (ER) and Unified Modeling Language (UML) models, followed by relational databases and querying them through SQL, relational design theory, indexing, query processing, transaction management, andbNoSQL/Big Data databases. In the web services part, the topics addressed include markup languages, three-tier application architectures, and web services. Security aspects are addressed from both perspectives.

A hands-on group project complements the theoretical aspects: on a self-chosen topic, students implement the core of a web-accessible information system using Python (or a similar language), MySQL, and Linux, guided through homework assignments.

Recommended Knowledge

Working knowledge of basic data structures, such as trees, is required as well as familiarity with an object-oriented programming language such as C++. Basic knowledge of algebra is useful. For the project work, students benefit from having basic hands-on skills using Linux and, ideally, basic knowledge of a scripting language such as Python (the official Python documentation is available on https://docs.python.org/).

Usability and Relationship to other Modules

This module introduces components that are widely used by modern applications and information systems. Students can apply their knowledge in the software engineering module. This module serves as a default advanced level minor module.

Intended Learning Outcomes

No	Competence	ILO
1	Read	Read and write ER and UML diagrams.
2	Design	Design and normalize data models for relational databases.
3	Write	Write SQL queries and understand their evaluation by a database server.
4	Explain	Explain the concept of transactions and how to use transactions in application design.
5	Use	Use web application frameworks to create dynamic websites.
6	Describe	Describe the differences of selected NoSQL data models and make a requirement-driven choice.
7	Restate	Restate three-tier architectures and their components.
8	Discuss	Discuss the principles and basic mechanisms of reactive website design.
9	Summarize	Summarize the security and privacy issues in the context of databases and web services.

Indicative Literature

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom: Database Systems: The Complete Book. 2nd edition, Pearson, 2008.
- Ragu Ramakrishnan: Database Management Systems. 3rd edition, McGraw Hill, 2003.
- James Lee: Open Source Web Development with LAMP. Pearson, 2003.

Entry Requirements

Prerequisites	Algorithms and Data Structures
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Databases and Web Services	Written Examination	120 minutes	67	45%	All intended learning outcomes of the excluding the practical aspects

Databases and Web Services - Project	Project Assessment		33	45%	All practical aspects of the intended learning outcomes

Module Achievements: None

6.9 Operating Systems

Module Name	Operating Systems
Module Code	2025-ACS-202
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 3 - 2025-S-ACS-BSc 4 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Jürgen Schönwälder

Forms of Learning and Teaching	
Asynchronous Self Study	52.5
Interactive Learning	57.5
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Operating Systems	ACS-202-A	Lecture (Online)	7.5

Module Description

This module introduces concepts and principles used by operating systems to provide programming abstractions that enable an efficient and robust execution of application programs. Students will gain an understanding of how an operating system kernel manages hardware components and how it provides abstractions such as processes, threads, virtual memory, file systems, and inter-process communication facilities. Students learn the principles of event-driven and concurrent programming and the mechanisms that are necessary to solve synchronization and coordination problems, thereby avoiding race conditions, deadlocks, and resource starvation. The Linux kernel and runtime system will be used throughout the course to illustrate how key ideas and concepts have been implemented and how application programs can use them.

Recommended Knowledge

Students are expected to have a working Linux installation, which allows them to compile and run sample programs provided by the instructor and to implement their own solutions for homework assignments.

Usability and Relationship to other Modules

This module enables students to write programs that make efficient use of the services provided by the operating system kernel. This is particularly important for advanced modules on computer networks, robotics, and embedded systems.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain the differences between processes, threads, application programs, libraries, and operating system kernels
2	Describe	Describe well-known mutual exclusion and coordination problems
3	Use	Use semaphores to achieve mutual exclusion and solve coordination problems
4	Use	Use mutual exclusion locks and condition variables to solve synchronization and coordination problems
5	Illustrate	Illustrate how deadlocks can be avoided, detected, and resolved
6	Summarize	Summarize the different mechanisms to realize virtual memory and their trade-offs
7	Solve	Solve basic inter-process communication problems using signals and pipes
8	Use	Use socket inter-process communication primitives
9	Multiplex	Multiplex I/O activities using suitable system calls and libraries
10	Describe	Describe file system programming interfaces and the design of file systems at the operating system kernel level
11	Explain	Explain how memory mapping can improve I/O performance
12	Restate	Restate the functionality of a linker and the difference between static linking and dynamic linking
13	Outline	Outline how different device types are supported by Unix-like kernels
14	Discuss	Discuss virtualization mechanisms such as containers or virtual machines

Indicative Literature

- Abraham Silberschatz, Peter B. Galvin, Greg Gagne: Applied Operating System Concepts, John Wiley, 2000.
- Andrew S. Tanenbaum, Herbert Bos: Modern Operating Systems, Prentice Hall, 4th edition, Pearson, 2015.
- William Stallings: Operating Systems: Internals and Design Principles, 8th edition, Pearson, 2014.
- Robert Love: Linux Kernel Development, 3rd edition, Addison Wesley, 2010.
- Robert Love: Linux System Programming: Talking Directly to the Kernel and C Library, 2nd edition, O'Reilly, 2013.

Entry Requirements

Prerequisites	Introduction to Computer Science Algorithms and Data Structures
Co-requisites	None
Additional Remarks	

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Operating Systems	Written Examination	120 minutes	100	45%	All intended learning outcomes of the module

Module Achievements: 50% of the assignments correctly solved. This module includes hands-on assignments so that students can develop their system programming skills. The module achievement ensures that a sufficient level of practical system programming skills has been obtained.

6.10 Probability and Random Processes

Module Name	Probability and Random Processes
Module Code	2025-CTMS-MAT-12-ONL
Module ECTS	5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 3 - 2025-S-ACS-BSc 4 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Keivan Mallahi Karai

Forms of Learning and Teaching	
Independent Study	90
Lecture	35
Workload Hours	125 hours

Module Components	Number	Type	CP
Probability and random processes	CTMS-12-ONL	Lecture	5

Module Description

This module aims to provide a basic knowledge of probability theory and random processes suitable for students in engineering, Computer Science, and Mathematics. The module provides students with basic skills needed for formulating real-world problems dealing with randomness and probability in mathematical language, and methods for applying a toolkit to solve these problems. Mathematical rigor is used where appropriate. A more advanced treatment of the subject is deferred to the third-year module Stochastic Processes.

The lecture comprises the following topics:

- Brief review of number systems, elementary functions, and their graphs

- Outcomes, events and sample space

- Combinatorial probability

- Conditional probability and Bayes’ formula

- Binomials and Poisson-Approximation

- Random Variables, distribution and density functions

- Independence of random variables

- Conditional Distributions and Densities

- Transformation of random variables

- Joint distribution of random variables and their transformations

- Expected Values and Moments, Covariance

- High dimensional probability: Chebyshev and Chernoff bounds

- Moment-Generating Functions and Characteristic Functions

- The Central limit theorem

- Random Vectors and Moments, Covariance matrix, Decorrelation

- Multivariate normal distribution. Markov chains, stationary distributions.

Recommended Knowledge

- Review all of the first-year calculus and linear algebra modules as indicated in "Entry Requirements - Knowledge, Ability, or Skills" above.

-Knowledge of calculus at the level of a first year calculus module (differentiation, integration with one and several variables, trigonometric functions, logarithms and exponential functions).

-Knowledge of linear algebra at the level of a first year university module (eigenvalues and eigenvectors, diagonalization of matrices).

- Some familiarity with elementary probability theory at the high school level.

Usability and Relationship to other Modules

Students taking this module are expected to be familiar with basic tools from calculus and linear algebra.

Intended Learning Outcomes

No	Competence	ILO
1	Command	Command the methods described in the content section of this module description to the extent that they can solve standard text-book problems reliably and with confidence.
2	Recognize	Recognize the probabilistic structures in an unfamiliar context and translate them into a mathematical problem statement.
3	Recognize	Recognize common mathematical terminology used in textbooks and research papers in the quantitative sciences, engineering, and mathematics to the extent that they fall into the content categories covered in this module.

Indicative Literature

- J. Hwang and J.K. Blitzstein (2019). Introduction to Probability, second edition. London: Chapman & Hall.
- S. Ghahramani. Fundamentals of Probability with Stochastic Processes, fourth edition. Upper Saddle River: Prentice Hall.

Entry Requirements

Prerequisites	Matrix Algebra and Advanced Calculus I Matrix Algebra and Advanced Calculus II
Co-requisites	None
Additional Remarks	

<u>Assessment and Completion</u>

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Probability and random processes	Written Examination	120 minutes	100	45%	1-3

Module Achievements: None

6.11 Data Analytics and Modeling

Module Name	Data Analytics and Modeling
Module Code	2025-CO-710
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-MDDA-BSc 3 - 2025-F-ACS-BSc 3 - 2025-S-ACS-BSc 2 - 2025-Minor-Data-Science 3 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-Minor-Data-Science (Minor in Data Science)
Module Coordinator(s)	Prof. Dr. Joachim Vogt Prof. Dr. Adalbert F.X. Wilhelm

Forms of Learning and Teaching	
Asynchronous Self Study	52.5
Interactive Learning	57.5
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Data Analytics and Modeling	CO-710-A	Lecture (Online)	7.5

Module Description

The module offers an introduction to the principles of data analytics and predictive data modeling and is structured into four parts. First, essential concepts from statistics are reviewed in the data modeling context, illustrating key ideas including randomness, distributions, and confidence regions. Examples and case studies are discussed to distinguish between proper and improper uses of statistics. Basic linear algebra is reviewed in the second part of the module, emphasizing vectors, distances, linear equations, matrices, and inversion. Key ideas such as the least squares approach are motivated with geometrical principles. The third part of the module is concerned with matrix decompositions such as the Singular Value Decomposition (SVD) and its close relatives Principal Component Analysis (PCA) and Empirical Orthogonal Function (EOF) analysis. The fourth part clarifies the distinction between linear and nonlinear modeling, and introduces key nonlinear techniques. Flexible educational formats (mostly online and hybrid) allow for asynchronous learning. Lectures are combined with Python exercises. Disciplinary applications and case studies are immersed as bridging elements.

Recommended Knowledge

Required for solving the coding assignments are Python skills at the level achieved after successful completion of the module Introduction to Data Science. Furthermore, students are encouraged to review first-year level statistics and linear algebra.

Intended Learning Outcomes

No	Competence	ILO
1	Identify	Identify important problem types and solution approaches in data analytics.
2	Understand	Understand how key concepts from statistics and linear algebra enter data science.
3	Explain	Explain matrix decompositions and their usage in data science.
4	Discuss	Discuss regularization concepts and optimality criteria in data analytics.
5	Know	Know the basics of nonlinear modeling and related computational approaches.
6	Convert	Convert data structures to Python/NumPy arrays for usage in data modeling.
7	Apply	Apply Python statistics and linear algebra tools in data analytics and modeling.

Indicative Literature

- Ani Adhikari, John DeNero, David Wagner. Computational and Inferential Thinking: The Foundations of Data Science 2019. Originally developed for the UC Berkeley course Data 8: Foundations of Data Science. An online version of the textbook is available at https://inferentialthinking.com/.
- Steven S. Skiena. The Data Science Design Manual. Springer 2017.
- Gilbert Strang: Linear Algebra and Learning from Data. Wellesley-Cambridge 2019. See https://math.mit.edu/~gs/learningfromdata/.
- Joe Suzuki: Statistical Learning with Math and Python. Springer 2021.
- Jake Vanderplas. Python Data Science Handbook. O'Reilly 2016. An online version is available at https://jakevdp.github.io/PythonDataScienceHandbook/.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Data Analytics and Modeling	Written Examination	180 minutes	100	45%	1-7

Module Achievements: 50% of the assignments need to be correctly solved.

6.12 Software Engineering

Module Name	Software Engineering
Module Code	2025-ACS-203
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-S-ACS-BSc 4 - 2025-F-ACS-BSc 4 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Peter Baumann

Forms of Learning and Teaching	
Asynchronous Self Study	35
Development Work	132.5
Exam Preparation	10
Independent Study	10
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Software Engineering Project	ACs-203-B	Project (Online)	5
Software Engineering	ACS-203-A	Lecture (Online)	2.5

Module Description

This module is an introduction to software engineering and object-oriented software design. The lecture focuses on software quality and the methods to achieve and maintain it in environments of "multi-person construction of multi-version software." Based on their pre-existing knowledge of an object-oriented programming language, students are familiarized with software architectures, design patterns and frameworks, software components and middleware, Unified Modeling Language (UML)-based modelling, and validation by testing. Furthermore, the course addresses the more organizational topics of project management and version control.

The lectures are accompanied by a software project in which students have to develop a software solution to a given problem. The problem is described from the viewpoint of a customer and students working in teams have to execute a whole software project lifecycle. The teams have to create a suitable software architecture and software design, implement the components, and integrate the components. The teams have to ensure that basic quality requirements for the solution and the components are defined and satisfied. The students produce various artifacts such as design documents, source code, test cases and user documentation. All artifacts need to be maintained in a version control system and the commits should allow the instructor and other team members to track in a meaningful way the changes and who has been contributing them.

Recommended Knowledge

Students are expected to be able to develop software using an object-oriented programming language such as C++, and they should have access to a Linux system and associated software development tools.

Intended Learning Outcomes

No	Competence	ILO
1	apply	apply tools that assist in the various stages of a software development process
2	assess	assess the quality of a software design and its implementation
3	contrast	contrast the benefits and drawbacks of different software development models
4	design	design and plan a larger software project involving a team development effort
5	evaluate	evaluate the applicability of different software engineering models for a given software development project
6	read	read and write UML diagrams
7	translate	translate requirements formulated by a customer into computer science terminology
8	understand	understand and apply object-oriented design patterns
9	work	work effectively in a team toward the goals of the team

Indicative Literature

- Ian Sommerville: Software Engineering Pearson 2010
- Roger Pressman: Software Engineering â€“ a Practitioner's Approach McGraw-Hill 2014

Entry Requirements

Prerequisites	Databases and Web Services
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Software Engineering Project	Project Assessment		67	45%	The remaining intended learning outcomes of the module (the project module component)

Software Engineering	Written Examination	60 minutes	33	45%	The first three intended learning outcomes of the module (the lecture module component)

Module Achievements: None

6.13 Artificial Intelligence

Module Name	Artificial Intelligence
Module Code	2025-ACS-204
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-S-ACS-BSc 3 - 2025-F-ACS-BSc 4 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Prof. Dr. Andreas Birk

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	17.5
Exam Preparation	20
Independent Study	115
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Artificial Intelligence	ACS-204-A	Lecture	5
Artificial Intelligence Tutorial	ACS-204-B	Tutorial	2.5

Module Description

Artificial Intelligence (AI) is an important subdiscipline of Computer Science that deals with technologies to automate the performance of tasks that are usually associated with intelligence. AI methods have a significant application potential, as there is an increasing interest and need to generate artificial systems that can carry out complex missions in unstructured environments without permanent human supervision. The module teaches a selection of the most important methods in AI. In addition to general-purpose techniques and algorithms, it also includes aspects of methods that are especially targeted for physical systems such as intelligent mobile robots or autonomous cars.

Usability and Relationship to other Modules

This module gives an introduction to Artificial Intelligence (AI) excluding the aspects of machine learning (ML), which are covered in a dedicated module that complements this one.

Recommended Knowledge

Revise content of the pre-requisite modules.

Intended Learning Outcomes

No	Competence	ILO
1	Outline	Outline and explain the history, general developments, and application areas of AI

2	Apply	Apply the basic concepts and methods of behavior-oriented AI
3	Use	Use concepts and methods of search algorithms for problem-solving
4	Explain	Explain the basic concepts of path-planning as an application example for domain-specific search
5	Apply	Apply basic path-planning algorithms and to compare their relations to general search algorithms
6	Write	Write and explain concepts of propositional and first-order logic
7	Use	Use logic representations and inference for basic examples of artificial planning systems
8	Apply	Apply AI concepts and methods to develop software

Indicative Literature

- S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.
- S. M. LaValle, Planning Algorithms. Cambridge University Press, 2006.
- J.-C. Latombe, Robot Motion Planning, Springer, 1991.

Entry Requirements

Prerequisites	Programming in C and C++ Introduction to RIS OR Introduction to CPS
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Artificial Intelligence	Written Examination	120 minutes	67	45%	All intended learning outcomes of the module excluding practical aspects
Artificial Intelligence Tutorial	Project Assessment		33	45%	All practical aspects of the intended learning outcomes

Module Achievements: None

6.14 Machine Learning

Module Name	Machine Learning
Module Code	2025-ACS-205
Module ECTS	7.5
Study Semester	Mandatory status for: - 2025-S-ACS-BSc 4 - 2025-F-ACS-BSc 4 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Prof. Dr. Andreas Birk

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	75
Exam Preparation	20
Independent Study	57.5
Workload Hours	187.5 hours

Module Components	Number	Type	CP
Machine Learning	ACS-205-A	Lecture (Online)	5
Machine Learning Tools	ACS-205-B	Laboratory	2.5

Module Description

Machine learning (ML) concerns algorithms that are fed with (large quantities of) real-world data, and which return a compressed “model” of the data. An example is the “world model” of a robot; the input data are sensor data streams, from which the robot learns a model of its environment, which is needed, for instance, for navigation. Another example is a spoken language model; the input data are speech recordings, from which ML methods build a model of spoken English; this is useful, for instance, in automated speech recognition systems. There exist many formalisms in which such models can be cast, and an equally large diversity of learning algorithms. However, there is a relatively small number of fundamental challenges that are common to all of these formalisms and algorithms. The lectures introduce such fundamental concepts and illustrate them with a choice of elementary model formalisms (linear classifiers and regressors, radial basis function networks, clustering, online adaptive filters, neural networks, or hidden Markov models). Furthermore, the lectures also (re-)introduce required mathematical material from probability theory and linear algebra. The ML lecture is complemented in this module by an online tutorial where the application-oriented side of software development in the context of ML is considered.

Modern machine learning in industry and research requires the knowledge of a comprehensive stack of tools and systems that allow to store and administrate data (e.g. Amazon S3, Kaggle, Dataverse, GIT LFS), extract features for various applications (e.g. Word2Vec, TSFEL), build up machine learning pipelines of training, testing, and hyperparameter optimization (e.g. skit-learn, Keras, TensorFlow, PyToarch) and ultimately deploy finalized models (e.g. TensorFlow Serving, MLFlow). This module gives exposure to a regularly updated latest state of the art set of tools that are relevant for the practical use of Machine Learning. It thereby complements the more theoretical and methods-driven module “Machine Learning” with market-oriented skills.

Recommended Knowledge

Knowledge and command of probability theory and methods, as in the module “Probability and Random Process" (CTMS-MAT-12)

Usability and Relationship to other Modules

This module gives a thorough introduction to the basics of machine learning. It complements the Artificial Intelligence module.

Intended Learning Outcomes

No	Competence	ILO
1	Understand	Understand the notion of probability spaces and random variables
2	Understand	Understand basic linear modeling and estimation techniques.
3	Understand	Understand the fundamental nature of the "curse of dimensionality".
4	Understand	Understand the fundamental nature of the bias-variance problem and standard coping strategies.
5	Use	Use elementary classification learning methods (linear discrimination, radial basis function networks, multilayer perceptrons).
6	Implement	Implement an end-to-end learning suite, including feature extraction and objective function optimization with regularization based on cross-validation.
7	Deploy	Deploy ML tools in an application context

Indicative Literature

- T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edition, Springer, 2008.
- S. Shalev-Shwartz, Shai Ben-David: Understanding Machine Learning, Cambridge University Press, 2014.
- C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- T.M. Mitchell, Machine Learning, Mc Graw Hill India, 2017.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Machine Learning	Written Examination	120 minutes	67	45%	All intended learning

					outcomes of the module excluding the practical aspects
Machine Learning Tools	Practical Assignments		33	45%	All intended learning outcomes of the module

Module Achievements: None

6.15 Computer Graphics

Module Name	Computer Graphics
Module Code	2025-ACS-303
Module ECTS	5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 5 - 2025-S-ACS-BSc 5 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Alexander Omelchenko

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	20
Exam Preparation	20
Independent Study	50
Workload Hours	125 hours

Module Components	Number	Type	CP
Computer Graphics	ACS-303-A	Lecture (Online)	5

Module Description

This module deals with the digital synthesis and manipulation of visual content. The creation process of computer graphics spans from the creation of a three-dimensional (3D) scene to displaying or storing it digitally. Prominent tasks in computer graphics are geometry processing, rendering, and animation. Geometry processing is concerned with object representations such as surfaces and their modeling. Rendering is concerned with transforming a model of the virtual world into a set of pixels by applying models of light propagation and sampling algorithms. Animation is concerned with descriptions of objects that move or deform over time. This is an introductory module covering the concepts and techniques of 3D (interactive) computer graphics. It covers mathematical foundations, basic algorithms and principles, and some advanced methods and concepts. An introduction to the implementation of simple programs using a mainstream computer graphics library completes this module.

Usability and Relationship to other Modules

Students with a strong interest in graphical user interfaces are encouraged to also select the Human–Computer Interaction specialization module, which discusses among other things how computer graphics can be used as a component of interactive graphical user interfaces.

Intended Learning Outcomes

No	Competence	ILO
1	Construct	Construct 3D geometry representations

2	Apply	Apply 3D transformations
3	Understand	Understand the algorithms and optimizations applied by graphics rendering systems
4	Explain	Explain the stages of modern computer graphics programmable pipelines
5	Implement	Implement simple computer graphics applications using graphics frameworks such as OpenGL
6	Illustrate	Illustrate the techniques used to create animations

Indicative Literature

- John Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, Kurt Akeley, Computer Graphics - Principles and Practice, 3rd edition, Addison-Wesley, 2013.
- Peter Shirley, Steve Marschner, Fundamentals of Computer Graphics, 4th edition, Taylor and Francis Ltd, 2016.
- Matt Pharr, Wenzel Jakob, Greg Humphreys, Physically Based Rendering: From Theory to Implementation, 3rd edition, Morgan Kaufmann, 2016.

Entry Requirements

Prerequisites	Algorithms and Data Structures
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Computer Graphics	Written Examination	120 minutes	100	45%	All intended learning outcomes of the module

Module Achievements: None

6.16 Computer Networks

Module Name	Computer Networks
Module Code	2025-ACS-304
Module ECTS	5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 3 - 2025-S-ACS-BSc 5 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Jürgen Schönwälder

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	20
Exam Preparation	20
Independent Study	50
Workload Hours	125 hours

Module Components	Number	Type	CP
Computer Networks	ACS-304-A	Lecture (Online)	5

Module Description

Computer networks such as the Internet play a critical role in today's connected world. This module discusses the technology of Internet services in depth to enable students to understand the core issues involved in the design of modern computer networks. Fundamental algorithms and principles are explained in the context of existing protocols as they are used in today's Internet. Students taking this course should finally understand the technical complexity behind every day online services such as Google or YouTube.

Students taking this module will understand how computer networks work and they will be able to assess communication networks, including aspects such as performance but also robustness and security. Students will learn that the design of communication networks is not only influenced by technical constraints but also by the necessity to define common standards, which often requires to take engineering decisions that reflect non-technical requirements.

Recommended Knowledge

Students are expected to be familiar with the C programming language and to learn basics of higher-level scripting languages such as Python (the official Python documentation is available on https://docs.python.org/).

Usability and Relationship to other Modules

The module should be taken together with the module Operating Systems, because a significant portion of the communication technology is implemented at the operating system level. An understanding of operating system concepts and abstractions will help students to understand how computer network technology is commonly implemented and made available to applications. The specialization module Distributed Algorithms discusses algorithms for solving problems commonly found in distributed systems that use computer networks to exchange information. The module Secure and Dependable Systems introduces cryptographic mechanisms that can be used to secure communication over computer networks.

Intended Learning Outcomes

No	Competence	ILO
1	Recall	Recall layering principles and the OSI reference model
2	Articulate	Articulate the organization of the Internet and the organization involved in providing Internet services
3	Describe	Describe media access control, flow control, and congestion control mechanisms
4	Explain	Explain how local area networks differ from global networks
5	Illustrate	Illustrate how frames are forwarded in local area networks
6	Contrast	Contrast addressing mechanisms and translations between addresses used at different layers
7	Demonstrate	Demonstrate how the Internet network layer forwards packets
8	Present	Present how routing algorithms and protocols are used to determine and select routes
9	Describe	Describe how the Internet transport layer provides different end-to-end services
10	Demonstrate	Demonstrate how names are resolved to addresses and vice versa
11	Summarize	Summarize how application layer protocols send and access electronic mail or access resources on the worldwide web
12	Design	Design and implement simple application layer protocols
13	Recognize	Recognize to which extent computer networks are fragile and evaluate strategies to cope with the fragility
14	Analyze	Analyze traffic traces produced by a given computer network

Indicative Literature

- James F. Kurose, Keith W. Ross: Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition, Addison-Wesley, 2004.
- Andrew S. Tanenbaum, Nick Feamster, David Wetherall: Computer Networks, 6th Edition, Pearson Education Limited, 2021.

Entry Requirements

Prerequisites	Algorithms and Data Structures
Co-requisites	Operating Systems
Additional Remarks	

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Computer Networks	Written Examination	120 minutes	100	45%	All intended learning outcomes of the module

Module Achievements: None

6.17 Web Application Development

Module Name	Web Application Development
Module Code	2025-ACS-305
Module ECTS	5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 5 - 2025-S-ACS-BSc 5 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CS-BSc (Computer Science)
Module Coordinator(s)	Prof. Dr. Alexander Omelchenko

Forms of Learning and Teaching	
Asynchronous Self Study	17.5
Interactive Learning	50
Exam Preparation	17.5
Independent Study	40
Workload Hours	125 hours

Module Components	Number	Type	CP
Web Application Development	ACS-305-A	Lecture (Online)	2.5
Web Application Development - Project	ACS-305-B	Project	2.5

Module Description

A web application is a client-server computer program where the client provides the user interface and the client side logic runs in a web browser or as an app running on a mobile device such as a smart phone or a tablet. A key characteristic is that more complex application logic and data storage is realized by a server offering a web application programming interface.

This module focuses on the client side of web application and introduces technologies that can be used to implement interactive user interfaces and client-side logic. It builds on the module databases and web services, which covers the data storage components and server-side logic of web applications.

This module consists of a lecture and an associated project. The lecture component introduces programming languages and frameworks that are widely used for implementing the client side of web applications such as Java, Kotlin, Swift, JavaScript and frameworks built on top of them. In the project component, students develop web applications and test them on existing and openly accessible web services.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain the document object model behind HTML and its relation to CSS
2	Discuss	Discuss the principles and basic mechanisms of reactive website design
3	Analyze	Analyze the interactions between web applications and web services

4	Use	Use languages such as Java, Kotlin, or Swift to implement mobile web applications
5	Use	Use web standards such as HTML, CSS, and JavaScript to implement web applications running in standard web browsers.

Indicative Literature

- Stoyan Stefanov: JavaScript Patterns, O'Reilly Media, 2010.
- Alexey Soshin: Hands-on Design Patterns with Kotlin, Packt Publishing, 2018.
- Alex Banks, Eve Porcello: Learning React: Functional Web Development.with React and Flux, O'Reilly, 2017.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Web Application Development	Written Examination	120 minutes	50	45%	First group of intended learning outcomes of the module
Web Application Development - Project	Project Assignment		50	45%	Second group of intended learning outcomes of the module

Module Achievements: None

6.18 Human Computer Interaction

Module Name	Human Computer Interaction
Module Code	2025-ACS-306
Module ECTS	5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 5 - 2025-S-ACS-BSc 5
Duration	1 Semester
Program Affiliation	2025-RIS-BSc (Robotics and Intelligent Systems)
Module Coordinator(s)	Prof. Dr. Francesco Maurelli

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	20
Exam Preparation	20
Independent Study	50
Workload Hours	125 hours

Module Components	Number	Type	CP
Human Computer Interaction	ACS-306-A	Lecture (Online)	5

Module Description

Computer systems often interact with human beings. The design of a good human–computer interface is often crucial for the acceptance and the success of a software system. Human–computer interface designs have to satisfy several requirements such as usability, learnability, efficiency, accessibility, and safety. The module discusses the evolution of human–computer interaction models and introduces design principles for graphical user interfaces and other types of interaction (e.g., visual, voice, gesture). Human–computer interaction designs are often evaluated using prototypes or mockups that can be given to test candidates to evaluate the effectiveness of the design. The module introduces evaluation strategies as well as tools and techniques that can be used to prototype human–computer interfaces.

Usability and Relationship to other Modules

Students with a strong interest in graphical user interfaces are encouraged to also select the Computer Graphics specialization module, which introduces methods and technologies for creating computer graphics and animations.

Intended Learning Outcomes

No	Competence	ILO
1	Explain	Explain the evolution of human–computer interaction models
2	Design	Design and implement simple graphical user interfaces
3	Explain	Explain ergonomic principles guiding the design of user interfaces

4	Illustrate	Illustrate different types of interaction (e.g., visual, voice, gestures) and their usability aspects
5	Evaluate	Evaluate aspects of and tradeoffs between usability, learnability, efficiency, and safety
6	Apply	Apply scientific methods to evaluate interfaces with respect to their usability and other desirable properties
7	Use	Use prototyping tools that can be employed to create mockups of user interfaces during the early stages of a software project

Indicative Literature

-

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Human Computer Interaction	Written Examination	120 minutes	100	45%	All intended learning outcomes of the module

Module Achievements: None

6.19 Collaborative Software Project

Module Name	Collaborative Software Project
Module Code	2025-ACS-301
Module ECTS	5
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 5 - 2025-S-ACS-BSc 6 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Prof. Dr. Andreas Birk

Forms of Learning and Teaching	
Asynchronous Self Study	5
Interactive Learning	60
Independent Study	60
Workload Hours	125 hours

Module Components	Number	Type	CP
Collaborative Software Project	ACS-301-A	Project (Online)	5

Module Description

The project enables the students to deepen their knowledge and skills in one or multiple areas of the 1st and especiallyb2nd year. They are exposed to state-of-the-art research with the goal to derive ideas and strategies to address application-oriented problems and to develop software for them. Students learn how to organize and execute an application-oriented research and development (R&D) project and how to present the results in the format of a white-paper. Students are expected to organize themselves in group work under the guidance of the instructor.

Intended Learning Outcomes

No	Competence	ILO
1	Understand	Understand state-of-the-art research papers in a chosen field of specialization
2	Plan	Plan a research project to reproduce research results or to extend ideas of recent research results
3	Explain	Explain research questions and choose suitable methodologies to address them
4	Use	Use methods and tools for remote collaborative software development
5	Document	Document a research project in the style of a typical white-paper

Indicative Literature

-

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	Students must have successfully passed 90 CP.

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Collaborative Software Project	Project Report	4,000 words	100	45%	All intended learning outcomes of the module

Module Achievements: None

6.20 Internship / Startup and Career Skills

Module Name	Internship / Startup and Career Skills
Module Code	2025-CA-INT-900
Module ECTS	15
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 5 - 2025-S-ACS-BSc 6 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	Career ()
Module Coordinator(s)	Dr. Tanja Woebs Clémentine Senicourt

Forms of Learning and Teaching	
Internship	308
Internship Event	2
Independent Study	32
Interactive Learning	33
Workload Hours	375 hours

Module Components	Number	Type	CP
Internship	CA-INT-900-0	Internship	15

Module Description

The aims of the internship module are reflection, application, orientation, and development: for students to reflect on their interests, knowledge, skills, their role in society, the relevance of their major subject to society, to apply these skills and this knowledge in real life whilst getting practical experience, to find a professional orientation, and to develop their personality and in their career. This module supports the programs' aims of preparing students for gainful, qualified employment and the development of their personality.

The full-time internship must be related to the students' major area of study and extends lasts a minimum of two consecutive months, normally scheduled just before the 5th semester, with the internship event and submission of the internship report in the 5th semester. Upon approval by the SPC and SCS, the internship may take place at other times, such as before teaching starts in the 3rd semester or after teaching finishes in the 6th semester. The Study Program Coordinator or their faculty delegate approves the intended internship a priori by reviewing the tasks in either the Internship Contract or Internship Confirmation from the respective internship institution or company. Further regulations as set out in the Policies for Bachelor Studies apply.

Students will be gradually prepared for the internship in semesters 1 to 4 through a series of mandatory information sessions, seminars, and career events.

The purpose of the Career Services Information Sessions is to provide all students with basic facts about the job market in general, and especially in Germany and the EU, and services provided by the Student Career Support.

In the Career Skills Seminars, students will learn how to engage in the internship/job search, how to create a competitive application (CV, Cover Letter, etc.), and how to successfully conduct themselves at job interviews and/or assessment centers. In addition to these mandatory sections, students can customize their skill set regarding application challenges and their intended career path in elective seminars.

Finally, during the Career Events organized by the Career Service Center (e.g. the annual Constructor Career Fair and single employer events on and off campus), students will have the opportunity to apply their acquired job market skills in an actual internship/job search situation and to gain their desired internship in a high-quality environment and with excellent employers.

As an alternative to the full-time internship, students can apply for the StartUp Option. Following the same schedule as the full-time internship, the StartUp Option allows students who are particularly interested in founding their own company to focus on the development of their business plan over a period of two consecutive months. Participation in the StartUp Option depends on a successful presentation of the student's initial StartUp idea. This presentation will be held at the beginning of the 4th semester. A jury of faculty members will judge the student's potential to realize their idea and approve the participation of the students. The StartUp Option is supervised by the Faculty StartUp Coordinator. At the end of StartUp Option, students submit their business plan. Further regulations as outlined in the Policies for Bachelor Studies apply.

The concluding Internship Event will be conducted within each study program (or a cluster of related study programs) and will formally conclude the module by providing students the opportunity to present on their internships and reflect on the lessons learned within their major area of study. The purpose of this event is not only to self-reflect on the whole internship process, but also to create a professional network within the academic community, especially by entering the Alumni Network after graduation. It is recommended that all three classes (years) of the same major are present at this event to enable networking between older and younger students and to create an educational environment for younger students to observe the "lessons learned" from the diverse internships of their elder fellow students.

Recommended Knowledge

- Information provided on CSC

- Major specific knowledge and skills

- Please see the section "Knowledge Center" at JobTeaser Career Center for information on Career Skills seminar and workshop offers and for online tutorials on the job market preparation and the application process. For more information, please see https://constructor.university/student-life/career-services

- Participating in the internship events of earlier classes

Usability and Relationship to other Modules

This module applies skills and knowledge acquired in previous modules to a professional environment and provides an opportunity to reflect on their relevance in employment and society. It may lead to thesis topics.

Intended Learning Outcomes

No	Competence	ILO
1	Describe	Describe the scope and the functions of the employment market and personal career development.
2	Apply	Apply professional, personal, and career-related skills for the modern labor market, including self-organization, initiative and responsibility, communication, intercultural sensitivity, team and leadership skills, etc.
3	Independently	Independently manage their own career orientation processes by identifying personal interests, selecting appropriate internship locations or start-up opportunities, conducting interviews, succeeding at pitches or assessment centers, negotiating related employment, managing their funding or support conditions (such as salary, contract, funding, supplies, work space, etc.).
4	Apply	Apply specialist skills and knowledge acquired during their studies to solve problems in a professional environment and reflect on their relevance in employment and society.
5	Justify	Justify professional decisions based on theoretical knowledge and academic methods.
6	Reflect	Reflect on their professional conduct in the context of the expectations of and consequences for employers and their society.
7	Reflect	Reflect on and set their own targets for the further development of their knowledge, skills, interests, and values.
8	Establish	Establish and expand their contacts with potential employers or business partners, and possibly other students and alumni, to build their own professional network to create employment opportunities in the future.
9	Discuss	Discuss observations and reflections in a professional network.

Indicative Literature

-

Entry Requirements

Prerequisites	Internship / Startup and Career Skills
Co-requisites	None
Additional Remarks	At least 15 CP from CORE modules in the major

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Internship	Project Report	3500 words	100	45%	1-9

Module Achievements: None

6.21 Bachelor Thesis ACS

Module Name	Bachelor Thesis ACS
Module Code	2025-ACS-400
Module ECTS	10
Study Semester	Mandatory status for: - 2025-F-ACS-BSc 6 - 2025-S-ACS-BSc 6 Mandatory Elective status for: None
Duration	14-week lecture period
Program Affiliation	2025-F-ACS-BSc (Applied Computer Science)
Module Coordinator(s)	Study Program Chair

Forms of Learning and Teaching	
Asynchronous Self Study	5
Interactive Learning	20
Independent Study/Laboratory Work	225
Workload Hours	250 hours

Module Components	Number	Type	CP
Thesis ACS	ACS-400-T	Thesis	10

Module Description

This module is a mandatory graduation requirement for all undergraduate students to demonstrate their ability to deal with a problem from their respective major subject independently by means of academic/scientific methods within a set period. Although supervised, the module requires the student to be able to work independently and regularly and set their own goals in exchange for the opportunity to explore a topic that excites and interests them personally and which a faculty member is interested to supervise. Within this module, students apply their acquired knowledge about the major discipline, skills, and methods to conduct research, ranging from the identification of suitable (short-term) research projects, preparatory literature searches, the realization of discipline-specific research, and the documentation, discussion, interpretation and communication of the results.

This module consists of an independent thesis. The thesis must be supervised by a Constructor University faculty member and requires short-term research work, the results of which must be documented in a comprehensive written thesis including an introduction, a justification of the methods, results, a discussion of the results, and conclusions.

Recommended Knowledge

- Comprehensive knowledge of the subject and deeper insight into the chosen topic

- Ability to plan and undertake work independently

- Skills to identify and critically review literature

- Identify an area or a topic of interest and discuss this with your prospective supervisor in good time

- Create a research proposal including a research plan to ensure timely submission

- Ensure you possess all required technical research skills or are able to acquire them on time

- Review again the University’s Code of Academic Integrity and Guidelines to Ensure Good Academic Practice

Usability and Relationship to other Modules

This module builds on all previous modules of the program. Students apply the knowledge, skills and competencies they acquired and practiced during their studies, including research methods and the ability to acquire additional skills independently as and if required.

Intended Learning Outcomes

No	Competence	ILO
1	Independently	Independently plan and organize advanced learning processes
2	Design	Design and implement appropriate research methods taking full account of the range of alternative techniques and approaches;
3	Collect	Collect, assess and interpret relevant information
4	Draw	Draw scientifically founded conclusions that consider social, scientific and ethical insights
5	Apply	Apply their knowledge and understanding to a context of their choice
6	Develop	Develop, formulate and advance solutions to problems and arguments in their subject area, and defend these through argument
7	Discuss	Discuss information, ideas, problems and solutions with specialists and non-specialists

Indicative Literature

-

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	Students must have taken and successfully passed a total of at least 30 CP from advanced modules, and of those at least 20 CP from advanced modules in the major. Non-major modules are the Management electives and New Skill modules.

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Thesis ACS	Thesis 80%, Presentation 20%	Presentation- 15 to 30 minutes		45%	All intended learning outcomes , mainly 1-

					6. The presentation focuses mainly on ILOs 6 and 7, but by nature of these ILOs also touches on the others.

Module Achievements: None

7 Management Modules

7.1 Digital Business Models and Functions

Module Name	Digital Business Models and Functions
Module Code	2025-MDSSB-DTRANS-02
Module ECTS	5
Study Semester	Mandatory status for: - 2025-DSSB-MSc 2 - 2025-MBA-120-MA 2 - 2025-MBA-60-MA 2 Mandatory Elective status for: - 2025-F-ACS-BSc 4 - 2025-S-ACS-BSc 3
Duration	1 Semester
Program Affiliation	2025-DSSB-MSc (Data Science for Society and Business)
Module Coordinator(s)	Prof. Dr. Tilo Halaszovich

Forms of Learning and Teaching	
Asynchronous Self Study	35
Interactive Learning	10
Exam Preparation	20
Independent Study	60
Workload Hours	125 hours

Module Components	Number	Type	CP
Digital Business Models and Functions	MDSSB-DTRANS-02	Lecture	5

Module Description

Businesses today have just begun to understand the potential of data abundance. Companies such as Amazon and Google were among the pioneers of data-driven business models. Many technology-based start-ups are eager to follow their lead. The data-driven revolution in the business world is nothing less than what Schumpeter termed a process of creative destruction. In this case, the destruction is of the long-established ways of doing business. The representatives of this new-age alternative business models range from shared economies and platform businesses to subscription models, even in the most traditional industries.

In this module, we will uncover the antecedents, drivers, and potentials of a data-driven economy by focusing on entrepreneurs and how their experiments creatively destruct the way we used to do business. We will explain why ecommerce is the fastest growing segment in retail today. We will examine e-commerce business models, technology infrastructure, e-commerce marketing and advertising concepts, social networks, auctions, and portals, as well as ethical, social, and political issues with the help of prominent case studies. At the end of the module, students will be able to build their own e-commerce (small-scale) companies.

Usability and Relationship to other Modules

This module focuses on digital business concepts and digital business models. It connects to all business modules in the "Society and Business" track to the core "Digital Transformation and Innovation" and "Artificial Intelligence in Business and Society" modules. However, it also forms the base for students who want to develop their own business ideas in the discovery section of the program and outside academia.

Recommended Knowledge

- Academic writing skills

- Good understanding of the principles of business functions

Intended Learning Outcomes

No	Competence	ILO
1	Know	Know about the development of business models on the Internet
2	Conceptually	Conceptually understand how to build an e-commerce presence
3	Comprehensively	Comprehensively understand e-commerce security and payment systems
4	Critically	Critically understand e-commerce marketing and advertising
5	Discuss	Discuss and reflect on major obstacles and possible solutions in e-commerce ethics
6	Critically	Critically evaluate and design business case studies

Indicative Literature

- Zott, Amit (2017) Business Model Innovation: How to Create Value in a Digital World. Marketing Intelligence Review 9 (1) DOI: https://doi.org/10.1515/gfkmir-2017-0003.
- Wirtz (2019) Digital Business Models: Concepts, Models, and the Alphabet Case Study. Cham: Springer Nature.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Digital Business Models and Functions	Term Paper	5000 words	100	45%	All

Module Achievements: None

7.2 Digital Marketing Fundamentals and SEO

Module Name	Digital Marketing Fundamentals and SEO
Module Code	2025-CTMS-MET-20
Module ECTS	5
Study Semester	Mandatory status for: - 2025-MDDA-BSc 3 Mandatory Elective status for: - 2025-IEM-BSc 3 - 2025-F-ACS-BSc 4 - 2025-S-ACS-BSc 3
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Dr. Matthias Meckel

Forms of Learning and Teaching	
Asynchronous Self Study	80
Independent Study	45
Workload Hours	125 hours

Module Components	Number	Type	CP
Digital Marketing Fundamentals and SEO	CTMS-20	Lecture	5

Module Description

This module is focused on key aspects of digital marketing and search engine optimisation (SEO), methodologies essential in today's online-driven marketing practice. State-of-the-art digital strategies, data-driven approaches, and SEO techniques will be at the core of the module.

The overall goal of this module is to help students without prior marketing knowledge to learn, understand, and practice the fundamentals of applied digital marketing methodology. This module helps students to navigate today's marketing challenges in a complex, dynamic online environment, where adaptability is essential, and where managers need to focus on achieving strategic goals through effective online presence rather than repetitive tasks.

Students will learn to develop and present consumer-centered and data-driven solutions for real-world digital marketing challenges.

Major challenges and concerns will be reflected:

- The role of data, the customer, and online visibility in a transformed digital landscape

- State-of-the-art digital marketing and SEO techniques

- Ethical and data security considerations in digital marketing

Intended Learning Outcomes

No	Competence	ILO

1	Develop	Develop practical knowledge and digital marketing skills, and mind sets to master address the challenges of today's online markets
2	Understand	Understand (routine) marketing processes in various context and how to state-of-the art digital methodologies to inform marketing decisions
3	Summarize	Summarize and classify the new data- and SEO-focused customer-driven methodologies within a digital marketing context
4	Understand	Understand the idea and potential for value -creation through of consumer-centric, data-informed digital marketingity
5	Apply	Apply innovative digital creativity methods and processes to enhance for marketing effectiveness

Indicative Literature

- Kotler, Keller, Chernev (2021): Marketing Management, Global Edition, 16th edition.
- Hanlon (2022): Digital Marketing, 2nd edition
- Charlesworth (2023): Digital Marketing A Practical Approach, 4th edition
- Dibb, Simkin, Pride Ferrell (2023): Marketing Concepts and Strategies, 9th edition
- Kotler (2022): Marketing: An Introduction, 15th edition
- Enge, Spencer, Stricchiola (2023): The Art of SEO, 4th edition

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Digital Marketing Fundamentals and SEO	Presentation	30 minutes	100	45%	All intended learning outcomes of the module

Module Achievements: None

8 Constructor Track Modules

8.1 Methods

8.1.1 Elements of Linear Algebra

Module Name	Elements of Linear Algebra
Module Code	2025-CTMS-MAT-24-ONL
Module ECTS	5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-RIS-BSc 1 - 2025-CS-BSc 1 - 2025-SDT-BSc 1 - 2025-F-ACS-BSc 1
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Keivan Mallahi Karai

Forms of Learning and Teaching	
Lecture	35
Independent Study	90
Workload Hours	125 hours

Module Components	Number	Type	CP
Elements of Linear Algebra	CTMS-24-ONL	Lecture	5

Module Description

This module is the first in a sequence introducing mathematical methods at the university level in a form relevant for study and research in the quantitative natural sciences, engineering, Computer Science. The emphasis in these modules is on training

operational skills and recognizing mathematical structures in a problem context. Mathematical rigor is used where appropriate. However, a full axiomatic treatment of the subject is provided in the first-year modules “Analysis” and “Linear Algebra”.

The lecture comprises the following topics:

- Review of elementary analytic geometry

- Vector spaces, linear independence, bases, coordinates

- Matrices and matrix algebra

- Solving linear systems by Gauss elimination, structure of general solution

- LU decomposition and matrix inverse

- Linear maps and connection to matrices

- Determinant

- Eigenvalues and eigenvectors

- Hermitian and skew-Hermitian matrices

- Orthonormal bases, Gram-Schmidt orthonormalization and QR decomposition

- Fourier transform

- Singular value decomposition

- Principal Component Analysis and best low rank approximations

Recommended Knowledge

- Knowledge of Pre-Calculus at High School level (Functions, inverse functions, sets, real numbers, trigonometric functions, parametric equations, tangent lines, graphs, elementary methods for solving systems of linear and nonlinear equations)

- Knowledge of Analytic Geometry at High School level (vectors, lines, planes, reflection, rotation, translation, dot product, cross product, normal vector, polar coordinates)

- Review all of higher-level High School Mathematics, in particular the topics explicitly named in "Entry Requirements – Knowledge, Ability, or Skills" above.

Usability and Relationship to other Modules

A rigorous treatment of this topic is provided in the module "Linear Algebra."

Intended Learning Outcomes

No	Competence	ILO
1	Apply	Apply the methods described in the content section of this module description to the extent that they can solve standard textbook problems reliably and with confidence.
2	Recognize	Recognize the mathematical structures in an unfamiliar context and translate them into a mathematical problem statement.
3	Recognize	Recognize common mathematical terminology and concepts used in textbooks and research papers in computer science, engineering, and mathematics to the extent that they fall into the content categories covered in this module.
4	Independently	Independently prove results which are direct consequences of those proved in the lectures
5	Understand	Understand and use fundamental mathematical terminology to communicate mathematical ideas.

Indicative Literature

- Gilbert Strang, Introduction to Linear Algebra, Fifth Edition (2016).
- S.A. Leduc Linear Algebra. Hoboken: Wiley (2003).

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Elements of Linear Algebra	Written Examination	120 minutes	100	45%	1-5

Module Achievements: None

8.1.2 Elements of Calculus

Module Name	Elements of Calculus
Module Code	2025-CTMS-MAT-25-ONL
Module ECTS	5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-RIS-BSc 2 - 2025-CS-BSc 2 - 2025-SDT-BSc 2 - 2025-F-ACS-BSc 2
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Keivan Mallahi Karai

Forms of Learning and Teaching	
Lecture	35
Independent Study	90
Workload Hours	125 hours

Module Components	Number	Type	CP
Elements of Calculus	CTMS-25-ONL	Lecture	5

Module Description

This module is the second in a sequence introducing mathematical methods at the university level in a form relevant for study and research in the quantitative natural sciences, engineering, Computer Science. The emphasis in these modules is on training operational skills and recognizing mathematical structures in a problem context. Mathematical rigor is used where appropriate. However, a full axiomatic treatment of the subject is provided in the first-year modules “Analysis”.

The lecture comprises the following topics:

- Sets, basic operations, and relations

- Functions, basic operations, compositions of functions, graphs of functions

- Brief introduction to real and complex numbers

- Limits for sequences and functions

- Continuity

- Derivatives of functions and its geometric interpretations

- Computing derivatives: linearity, product rule, chain rule

- Applications of derivatives, optimization for one-variable functions

- Introduction to Integration and the Fundamental Theorem of Calculus

- Differential equations, modeling simple dynamical systems

- Discrete derivative, summations, difference equations

- Functions of several variables, representations using graphs and level curves

- Basic ideas of multivariable calculus

- Partial derivatives and directional derivatives, total derivative

- Optimization in several variables, gradient descent, Lagrange multipliers

- Ordinary differential equations with several variables, simple examples

- Linear constant-coefficient ordinary differential equations

- Fourier series and their applications

Recommended Knowledge

- Knowledge of Pre-Calculus at High School level (Functions, inverse functions, sets, real numbers, polynomials, rational functions, trigonometric functions, logarithm and exponential function, parametric equations, tangent lines, graphs.)

- Knowledge of Analytic Geometry at High School level (vectors, lines, planes, reflection, rotation, translation, dot product, cross product, normal vector, polar coordinates)

- Some familiarity with elementary Calculus (limits, derivative) is helpful, but not strictly required.

- Review the content of Linear Algebra

Usability and Relationship to other Modules

A rigorous treatment of this topic is provided in the module "Analysis"

Intended Learning Outcomes

No	Competence	ILO
1	Apply	Apply the methods described in the content section of this module description to the extent that they can solve standard textbook problems reliably and with confidence.
2	Recognize	Recognize the mathematical structures in an unfamiliar context and translate them into a mathematical problem statement.
3	Recognize	Recognize common mathematical terminology and concepts used in textbooks and research papers in computer science, engineering, and mathematics to the extent that they fall into the content categories covered in this module.
4	Independently	Independently prove results which are direct consequences of those proved in the lectures.
5	Understand	Understand and use fundamental mathematical terminology to communicate mathematical ideas.

Indicative Literature

- James Stewart, Calculus: Early Transcendentals, (2015).

- S.I. Grossman, Calculus of one variable, 2nd edition, (2014).

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Elements of Calculus	Written Examination	120 minutes	100	45%	1-5

Module Achievements: None

8.1.3 Probability and Random Processes

Module Name	**Probability and Random Processes**
Module Code	2025-CTMS-MAT-12
Module ECTS	5
Study Semester	Mandatory status for: - 2025-PHDS-BSc 3 - 2025-RIS-BSc 3 - 2025-ECE-BSc 3 - 2025-MMDA-BSc 3 - 2025-CS-BSc 3 - 2025-SDT-BSc 3 - 2025-F-ACS-BSc 3 - 2025-S-ACS-BSc 4 Mandatory Elective status for: None
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Keivan Mallahi Karai

Forms of Learning and Teaching	
Independent Study	90
Lecture	35
Workload Hours	125 hours

Module Components	**Number**	**Type**	**CP**
Probability and random processes	CTMS-12	Lecture	5

Module Description

This module aims to provide a basic knowledge of probability theory and random processes suitable for students in engineering, Computer Science, and Mathematics. The module provides students with basic skills needed for formulating real-world problems dealing with randomness and probability in mathematical language, and methods for applying a toolkit to solve these problems. Mathematical rigor is used where appropriate. A more advanced treatment of the subject is deferred to the third-year module Stochastic Processes.

The lecture comprises the following topics:

- Brief review of number systems, elementary functions, and their graphs

- Outcomes, events and sample space

- Combinatorial probability

- Conditional probability and Bayes’ formula

- Binomials and Poisson-Approximation

- Random Variables, distribution and density functions

- Independence of random variables

- Conditional Distributions and Densities

- Transformation of random variables

- Joint distribution of random variables and their transformations

- Expected Values and Moments, Covariance

- High dimensional probability: Chebyshev and Chernoff bounds

- Moment-Generating Functions and Characteristic Functions

- The Central limit theorem

- Random Vectors and Moments, Covariance matrix, Decorrelation

- Multivariate normal distribution. Markov chains, stationary distributions.

Recommended Knowledge

- Review all of the first-year calculus and linear algebra modules as indicated in "Entry Requirements - Knowledge, Ability, or Skills" above.

-Knowledge of calculus at the level of a first year calculus module (differentiation, integration with one and several variables, trigonometric functions, logarithms and exponential functions).

-Knowledge of linear algebra at the level of a first year university module (eigenvalues and eigenvectors, diagonalization of matrices).

- Some familiarity with elementary probability theory at the high school level.

Usability and Relationship to other Modules

Students taking this module are expected to be familiar with basic tools from calculus and linear algebra.

Intended Learning Outcomes

No	Competence	ILO
1	Command	Command the methods described in the content section of this module description to the extent that they can solve standard text-book problems reliably and with confidence.
2	Recognize	Recognize the probabilistic structures in an unfamiliar context and translate them into a mathematical problem statement.
3	Recognize	Recognize common mathematical terminology used in textbooks and research papers in the quantitative sciences, engineering, and mathematics to the extent that they fall into the content categories covered in this module.

Indicative Literature

- J. Hwang and J.K. Blitzstein (2019). Introduction to Probability, second edition. London: Chapman & Hall.

- S. Ghahramani. Fundamentals of Probability with Stochastic Processes, fourth edition. Upper Saddle River: Prentice Hall.

Entry Requirements

Prerequisites	Elements of Linear Algebra Elements of Calculus Matrix Algebra and Advanced Calculus I Matrix Algebra and Advanced Calculus II
Co-requisites	None
Additional Remarks	

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Probability and random processes	Written Examination	120 minutes	100	45%	1-3

Module Achievements: None

8.2 New Skills

8.2.1 Logic (perspective I)

Module Name	Logic (perspective I)
Module Code	2025-CTNS-NSK-01
Module ECTS	2.5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 3
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Jules Coleman

Forms of Learning and Teaching	
Independent Study	45
Online Lecture	17.5
Workload Hours	62.5 hours

Module Components	Number	Type	CP
Logic (perspective I)	CTNS-01	Lecture (Online)	2.5

Module Description

Suppose a friend asks you to help solve a complicated problem? Where do you begin? Arguably, the first and most difficult task you face is to figure out what the heart of the problem actually is. In doing that you will look for structural similarities between the problem posed and other problems that arise in different fields that others may have addressed successfully. Those similarities may point you to a pathway for resolving the problem you have been asked to solve. But it is not enough to look for structural similarities. Sometimes relying on similarities may even be misleading. Once you've settled tentatively on what you take to be the heart of the matter, you will naturally look for materials, whether evidence or arguments, that you believe is relevant to its potential solution. But the evidence you investigate of course depends on your formulation of the problem, and your formulation of the problem likely depends on the tools you have available - including potential sources of evidence and argumentation. You cannot ignore this interactivity, but you can't allow yourself to be hamstrung entirely by it. But there is more. The problem itself may be too big to be manageable all at once, so you will have to explore whether it can be broken into manageable parts and if the information you have bears on all or only some of those parts. And later you will face the problem of whether the solutions to the particular sub problems can be put together coherently to solve the entire problem taken as a whole.

What you are doing is what we call engaging in computational thinking. There are several elements of computational thinking illustrated above. These include: Decomposition (breaking the larger problem down into smaller ones); Pattern recognition (identifying structural similarities); Abstraction (ignoring irrelevant particulars of the problem): and Creating Algorithms), problem-solving formulas.

But even more basic to what you are doing is the process of drawing inferences from the material you have. After all, how else are you going to create a problem-solving formula, if you draw incorrect inferences about what information has shown and what, if anything follows logically from it. What you must do is apply the rules of logic to the information to draw inferences that are warranted.

We distinguish between informal and formal systems of logic, both of which are designed to indicate fallacies as well as warranted inferences. If I argue for a conclusion by appealing to my physical ability to coerce you, I prove nothing about the truth of what I claim. If anything, by doing so I display my lack of confidence in my argument. Or if the best I can do is berate you for your skepticism, I have done little more than offer an ad hominem instead of an argument. Our focus will be on formal systems of logic, since they are at the heart of both scientific argumentation and computer developed algorithms. There are in fact many different kinds of logic and all figure to varying degrees in scientific inquiry. There are inductive types of logic, which purport to formalize the relationship between premises that if true offer evidence on behalf of a conclusion and the conclusion and are represented as claims about the extent to which the conclusion is confirmed by the premises. There are deductive types of logic, which introduce a different relationship between premise and conclusion. These variations of logic consist in rules that if followed entail that if the premises are true then the conclusion too must be true.

There are also modal types of logic which are applied specifically to the concepts of necessity and possibility, and thus to the relationship among sentences that include either or both those terms. And there is also what are called deontic logic, a modification of logic that purport to show that there are rules of inference that allow us to infer what we ought to do from facts about the circumstances in which we find ourselves. In the natural and social sciences most of the emphasis has been placed on inductive logic, whereas in math it is placed on deductive logic, and in modern physics there is an increasing interest in the concepts of possibility and necessity and thus in modal logic. The humanities, especially normative discussions in philosophy and literature are the province of deontic logic.

This module will also take students through the central aspects of computational thinking, as it is related to logic; it will introduce the central concepts in each, their relationship to one another and begin to provide the conceptual apparatus and practical skills for scientific inquiry and research.

Intended Learning Outcomes

No	Competence	ILO
1	Apply	Apply the various principles of logic and expand them to computational thinking.
2	Understand	Understand the way in which logical processes in humans and in computers are similar and different at the same time.
3	Apply	Apply the basic rules of first-order deductive logic and employ them rules in the context of creating a scientific or social scientific study and argument.
4	Employ	Employ those rules in the context of creating a scientific or social scientific study and argument.

Indicative Literature

- Frege, Gottlob (1879), Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens [Translation: A Formal Language for Pure Thought Modeled on that of Arithmetic],Halle an der Salle: Verlag von Louis Nebert.
- Gödel, Kurt (1986), Russels mathematische Logik. In: Alfred North Whitehead, Bertrand Russell: Principia Mathematica. Vorwort, S. V–XXXIV. Suhrkamp.
- Leeds, Stephen. "George Boolos and Richard Jeffrey. Computability and logic. Cambridge University Press, New York and London1974, x+ 262 pp." The Journal of Symbolic Logic 42.4 (1977): 585-586.
- Kubica, Jeremy. Computational fairy tales. Jeremy Kubica, 2012.
- McCarthy, Timothy. "Richard Jeffrey. Formal logic: Its scope and limits. of XXXVIII 646. McGraw-Hill Book Company, New York etc. 1981, xvi+ 198 pp." The Journal of Symbolic Logic 49.4 (1984): 1408-1409.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Logic (perspective I)	Written Examination	60 minutes	100	45%	All

Module Achievements: None

8.2.2 Logic (perspective II)

Module Name	Logic (perspective II)
Module Code	2025-CTNS-NSK-02
Module ECTS	2.5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 3
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Jules Coleman

Forms of Learning and Teaching	
Independent Study	45
Online Lecture	17.5
Workload Hours	62.5 hours

Module Components	Number	Type	CP
Logic (perspective II)	CTNS-02	Lecture (Online)	2.5

Module Description

The focus of this module is on formal systems of logic, since they are at the heart of both scientific argumentation and computer developed algorithms. There are in fact many kinds of logic and all figure to varying degrees in scientific inquiry. There are inductive types of logic, which purport to formalize the relationship between premises that if true offer evidence on behalf of a conclusion and the conclusion and are represented as claims about the extent to which the conclusion is confirmed by the premises. There are deductive types of logic, which introduce a different relationship between premise and conclusion. These variations of logic consist in rules that if followed entail that if the premises are true then the conclusion too must be true.

This module introduces logics that go beyond traditional deductive propositional logic and predicate logic and as such it is aimed at students who are already familiar with basics of traditional formal logic. The aim of the module is to provide an overview of alternative logics and to develop a sensitivity that there are many different logics that can provide effective tools for solving problems in specific application domains.

The module first reviews the principles of a traditional logic and then introduces many-valued logics that distinguish more than two truth values, for example true, false, and unknown. Fuzzy logic extends traditional logic by replacing truth values with real numbers in the range 0 to 1 that are expressing how strong the believe into a proposition is. Modal logics introduce modal operators expressing whether a proposition is necessary or possible. Temporal logics deal with propositions that are qualified by time. One can view temporal logics as a form of modal logics where propositions are qualified by time constraints. Interval temporal logic provides a way to reason about time intervals in which propositions are true.

The module will also investigate the application of logic frameworks to specific classes of problems. For example, a special subset of predicate logic, based on so-called Horn clauses, forms the basis of logic programming languages such as Prolog. Description logics, which are usually decidable logics, are used to model relationships and they have applications in the semantic web, which enables search engines to reason about resources present on the Internet.

Intended Learning Outcomes

No	Competence	ILO
1	Apply	Apply the various principles of logic.
2	Explain	Explain practical relevance of non-standard logic.
3	Describe	Describe how many-valued logic extends basic predicate logic.
4	Apply	Apply basic rules of fuzzy logic to calculate partial truth values.
5	Sketch	Sketch basic rules of temporal logic.
6	Implement	Implement predicates in a logic programming language.
7	Prove	Prove some simple non-standard logic theorems.

Indicative Literature

- Bergmann, Merry. “An Introduction to Many-Valued and Fuzzy Logic: Semantics, Algebras, and Derivation Systems”, Cambridge University Press, April 2008.
- Sterling, Leon S., Ehud Y. Shapiro, Ehud Y. "The Art of Prolog", 2nd edition, MIT Press, March 1994.
- Fisher, Michael. "An Introduction to Practical Formal Methods Using Temporal Logic", Wiley, Juli 2011.
- Baader, Franz. "The Description Logic Handbook: Theory Implementation and Applications", Cambridge University Press, 2nd edition, May 2010.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Logic (perspective II)	Written Examination	60 minutes	100	45%	All

Module Achievements: None

8.2.3 Causation and Correlation (perspective I)

Module Name	**Causation and Correlation (perspective I)**
Module Code	2025-CTNS-NSK-03
Module ECTS	2.5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 4
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Jules Coleman

Forms of Learning and Teaching	
Independent Study	45
Online Lecture	17.5
Workload Hours	62.5 hours

Module Components	**Number**	**Type**	**CP**
Causation and Correlation	CTNS-03	Lecture (Online)	2.5

Module Description

In many ways, life is a journey. And also, as in other journeys, our success or failure depends not only on our personal traits and character, our physical and mental health, but also on the accuracy of our map. We need to know what the world we are navigating is actually like, the how, why and the what of what makes it work the way it does. The natural sciences provide the most important tool we have developed to learn how the world works and why it works the way it does. The social sciences provide the most advanced tools we have to learn how we and other human beings, similar in most ways, different in many others, act and react and what makes them do what they do. In order for our maps to be useful, they must be accurate and correctly reflect the way the natural and social worlds work and why they work as they do.

The natural sciences and social sciences are blessed with enormous amounts of data. In this way, history and the present are gifts to us. To understand how and why the world works the way it does requires that we are able to offer an explanation of it. The data supports a number of possible explanations of it. How are we to choose among potential explanations? Explanations, if sound, will enable us to make reliable predictions about what the future will be like, and also to identify many possibilities that may unfold in the future. But there are differences not just in the degree of confidence we have in our predictions, but in whether some of them are necessary future states or whether all of them are merely possibilities? Thus, there are three related activities at the core of scientific inquiry: understanding where we are now and how we got here (historical); knowing what to expect going forward (prediction); and exploring how we can change the paths we are on (creativity).

At the heart of these activities are certain fundamental concepts, all of which are related to the scientific quest to uncover immutable and unchanging laws of nature. Laws of nature are thought to reflect a causal nexus between a previous event and a future one. There are also true statements that reflect universal or nearly universal connections between events past and present that are not laws of nature because the relationship they express is that of a correlation between events. A working thermostat accurately allows us to determine or even to predict the temperature in the room in which it is located, but it does not explain why the room has the temperature it has. What then is the core difference between causal relationships and correlations? At the same time, we all recognize that given where we are now there are many possible futures for each of us, and even had our lives gone just the slightest bit differently than they have, our present state could well have been very different than it is. The relationship between possible pathways between events that have not materialized but could have is expressed through the idea of counterfactual.

Creating accurate roadmaps, forming expectations we can rely on, making the world a more verdant and attractive place requires us to understand the concepts of causation, correlation, counterfactual explanation, prediction, necessity, possibility, law of nature and universal generalization. This course is designed precisely to provide the conceptual tools and intellectual skills to implement those concepts in our future readings and research and ultimately in our experimental investigations, and to employ those tools in various disciplines.

Intended Learning Outcomes

No	Competence	ILO
1	Formulate	Formulate testable hypotheses that are designed to reveal causal connections and those designed to reveal interesting, important and useful correlations.
2	Distinguish	Distinguish scientifically interesting correlations from unimportant ones.
3	Apply	Apply critical thinking skills to evaluate information.
4	Understand	Understand when and why inquiry into unrealized possibility is important and relevant.

Indicative Literature

- Thomas S. Kuhn: The Structure of Scientific Revolutions. Nelson, fourth edition, 2012.
- Goodman, Nelson. Fact, fiction, and forecast. Harvard University Press, 1983.
- Quine Willard, Van Orman, and Joseph Silbert Ullian. The web of belief. Vol 2. New York: Random house, 1978.

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Causation and Correlation	Written Examination	60 minutes	100	45%	1-4

Module Achievements: None

8.2.4 Causation and Correlation (perspective II)

Module Name	**Causation and Correlation (perspective II)**
Module Code	2025-CTNS-NSK-04
Module ECTS	2.5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 4 - 2025-S-ACS-BSc 3
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Dr. Eoin Ryan Dr. Irina Chiaburu Prof. Dr. Keivan Mallahi Karai

Forms of Learning and Teaching	
Independent Study	45
Online Lecture	17.5
Workload Hours	62.5 hours

Module Components	**Number**	**Type**	**CP**
Causation and Correlation (perspective II)	CTNS-04	Lecture (Online)	2.5

Module Description

Causality or causation is a surprisingly difficult concept to understand. David Hume famously noted that causality is a concept that our science and philosophy cannot do without, but it is equally a concept that our science and philosophy cannot describe. Since Hume, the problem of cause has not gone away, and sometimes seems to get even worse (e.g., quantum mechanics confusing previous notions of causality). Yet, ways of doing science that lessen our need to explicitly use causality have become very effective (e.g., huge developments in statistics). Nevertheless, it still seems that the concept of causality is at the core of explaining how the world works, across fields as diverse as physics, medicine, logistics, the law, sociology, and history - and ordinary daily life - through all of which, explanations and predictions in terms of cause and effect remain intuitively central.

Causality remains a thorny problem but, in recent decades, significant progress has occurred, particularly in work by or inspired by Judea Pearl. This work incorporates many 20th century developments, including statistical methods - but with a reemphasis on finding the why, or the cause, behind statistical correlations -, progress in understanding the logic, semantics and metaphysics of conditionals and counterfactuals, developments based on insights from the likes of philosopher Hans Reichenbach or biological statistician Sewall Wright into causal precedence and path analysis, and much more. The result is a new toolkit to identify causes and build causal explanations. Yet even as we get better at identifying causes, this raises new (or old) questions about causality, including metaphysical questions about the nature of causes (and effects, events, objects, etc), but also questions about what we really use causality for (understanding the world as it is or just to glean predictive control of specific outcomes), about how causality is used differently in different fields and activities (is cause in physics the same as that in history?), and about how other crucial concepts relate to our concept of cause (space and time seem to be related to causality, but so do concepts of legal and moral responsibility).

This course will introduce students to the mathematical formalism derived from Pearl's work, based on directed acyclic graphs and probability theory. Building upon previous work by Reichenbach and Wright, Pearl defines a "a calculus of interventions" of "do-calculus" for talking about interventions and their relation to causation and counterfactuals. This model has been applied in various areas ranging from econometrics to statistics, where acquiring knowledge about causality is of great importance.

At the same time, the course will not forget some of the metaphysical and epistemological issues around cause, so that students can better critically evaluate putative causal explanations in their full context. Abstractly, such issues involve some of the same philosophical questions Hume already asked, but more practically, it is important to see how metaphysical and epistemological debates surrounding the notion of cause affect scientific practice, and equally if not more importantly, how scientific practice pushes the limits of theory. This course will look at various ways in which empirical data can be transformed into explanations and theories, including the variance approach to causality (characteristic of the positivistic quantitative paradigm), and the process theory of causality (associated with qualitative methodology). Examples and case studies will be relevant for students of the social sciences but also students of the natural/physical world as well.

Recommended Knowledge

Basic probability theory

Intended Learning Outcomes

No	Competence	ILO
1	Have	Have a clear understanding of the history of causal thinking.
2	Form	Form a critical understanding of the key debates and controversies surrounding the idea of causality.
3	Recognize	Recognize and apply probabilistic causal models.
4	Explain	Explain how understanding of causality differs among different disciplines.
5	Demonstrate	Demonstrate how theoretical thinking about causality has shaped scientific practices.

Indicative Literature

- Paul, L. A. and Ned Hall. Causation: A User’s Guide. Oxford University Press 2013.
- Pearl, Judea. Causality: Models, Reasoning and Inference. Cambridge University Press 2009.
- Pearl, Judea, Glymour Madelyn and Jewell, Nicolas. Causal Inference in Statistics: A Primer. Wiley 2016.
- llari, Phyllis McKay and Federica Russo. Causality: Philosophical Theory Meets Scientific Practice. Oxford University Press 2014.

Entry Requirements

Prerequisites	None

Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Causation and Correlation (perspective II)	Written Examination	60 minutes	100	45%	1-5

Module Achievements: None

8.2.5 Argumentation, Data Visualization and Communication (perspective I)

Module Name	**Argumentation, Data Visualization and Communication (perspective I)**
Module Code	2025-CTNS-NSK-07
Module ECTS	5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 5 - 2025-S-ACS-BSc 5
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Arvid Kappas Prof. Dr. Jules Coleman

Forms of Learning and Teaching	
Independent Study	90
Online Lecture	35
Workload Hours	125 hours

Module Components	**Number**	**Type**	**CP**
Argumentation, Data Visualization and Communication (perspective I)	CTNS-07	Lecture (Online)	5

Module Description

One must be careful not to confuse argumentation with being argumentative. The latter is an unattractive personal attribute, whereas the former is a requirement of publicly holding a belief, asserting the truth of a proposition, the plausibility of a hypothesis, or a judgment of the value of a person or an asset. It is an essential component of public discourse. Public discourse is governed by norms and one of those norms is that those who assert the truth of a proposition or the validity of an argument or the responsibility of another for wrongdoing open themselves up to good faith requests to defend their claims. In its most general meaning, argumentation is the requirement that one offer evidence in support of the claims they make, as well as in defense of the judgments and assessments they reach. There are different modalities of argumentation associated with different contexts and disciplines. Legal arguments have a structure of their own as do assessments of medical conditions and moral character. In each case, there are differences in the kind of evidence that is thought relevant and, more importantly, in the standards of assessment for whether a case has been successfully made. Different modalities of argumentation require can call for different modes of reasoning. We not only offer reasons in defense of or in support of beliefs we have, judgments we make and hypotheses we offer, but we reason from evidence we collect to conclusions that are warranted by them.

Reasoning can be informal and sometimes even appear unstructured. When we recognize some reasoning as unstructured yet appropriate what we usually have in mind is that it is not linear. Most reasoning we are familiar with is linear in character. From A we infer B, and from A and B we infer C, which all together support our commitment to D. The same form of reasoning applies whether the evidence for A, B or C is direct or circumstantial. What changes in these cases is perhaps the weight we give to the evidence and thus the confidence we have in drawing inferences from it.

Especially in cases where reasoning can be supported by quantitative data, wherever quantitative data can be obtained either directly or by linear or nonlinear models, the visualization of the corresponding data can become key in both, reasoning and argumentation. A graphical representation can reduce the complexity of argumentation and is considered a must in effective scientific communication. Consequently, the course will also focus on smart and compelling ways for data visualization - in ways that go beyond what is typically taught in statistics or mathematics lectures. These tools are constantly developing, as a reflection of new software and changes in state of the presentation art. Which graph or bar chart to use best for which data, the use of colors to underline messages and arguments, but also the pitfalls when presenting data in a poor or even misleading manner. This will also help in readily identifying intentional mis-representation of data by others, the simplest to recognize being truncating the ordinate of a graph in order to exaggerate trends. This frequently leads to false arguments, which can then be readily countered.

There are other modalities of reasoning that are not linear however. Instead they are coherentist. We argue for the plausibility of a claim sometimes by showing that it fits in with a set of other claims for which we have independent support. The fit is itself the reason that is supposed to provide confidence or grounds for believing the contested claim.

Other times, the nature of reasoning involves establishing not just the fit but the mutual support individual items in the evidentiary set provide for one another. This is the familiar idea of a web of interconnected, mutually supportive beliefs. In some cases, the support is in all instances strong; in others it is uniformly weak, but the set is very large; in other cases, the support provided each bit of evidence for the other is mixed: sometimes strong, sometimes weak, and so on.

There are three fundamental ideas that we want to extract from this segment of the course. These are (1) that argumentation is itself a requirement of being a researcher who claims to have made findings of one sort or another; (2) that there are different forms of appropriate argumentation for different domains and circumstances; and (3) that there are different forms of reasoning on behalf of various claims or from various bits of evidence to conclusions: whether those conclusions are value judgments, political beliefs, or scientific conclusions. Our goal is to familiarize you with all three of these deep ideas and to help you gain facility with each.

Intended Learning Outcomes

No	Competence	ILO
1	Distinguish	Distinguish among different modalities of argument, e.g. legal arguments, vs. scientific ones.
2	Construct	Construct arguments using tools of data visualization.
3	Communicate	Communicate conclusions and arguments concisely, clearly and convincingly.

Indicative Literature

- Tufte, E.R. (1985). The visual display of quantitative information. The Journal for Healthcare Quality (JHQ), 7(3), 15.
- Cairo, A (2012). The Functional Art: An introduction to information graphics and visualization. New Ridders.

- Knaflic, C.N. (2015). Storytelling with data: A data visualization guide for business professionals. John Wiley & Sons.

Entry Requirements

Prerequisites	Logic (perspective I) Causation and Correlation (perspective I) Causation and Correlation (perspective II) Logic (perspective II)
Co-requisites	None
Additional Remarks	

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Argumentation, Data Visualization and Communication (perspective I)	Written Examination	120 minutes	100	45%	1-3

Module Achievements: None

8.2.6 Argumentation, Data Visualization and Communication (perspective II)

Module Name	**Argumentation, Data Visualization and Communication (perspective II)**
Module Code	2025-CTNS-NSK-08
Module ECTS	5
Study Semester	Mandatory status for: None Mandatory Elective status for: - 2025-F-ACS-BSc 6 - 2025-S-ACS-BSc 6
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Arvid Kappas Prof. Dr. Jules Coleman

Forms of Learning and Teaching	
Independent Study	80
Online Lecture	35
Tutorial	10
Workload Hours	125 hours

Module Components	**Number**	**Type**	**CP**
Argumentation, Data Visualization and Communication (perspective II)	CTNS-08	Lecture (Online)	5

Module Description

Humans are a social species, and interaction is crucial throughout the entire life span. While much of human communication involves language, there is a complex multichannel system of nonverbal communication that enriches linguistic content, provides context, and is also involved in structuring dynamic interaction. Interactants achieve goals by encoding information that is interpreted in the light of current context in transactions with others. This complexity implies also that there are frequent misunderstandings as a sender's intention is not fulfilled. Students in this course will learn to understand the structure of communication processes in a variety of formal and informal contexts. They will learn what constitutes challenges to achieving successful communication and to how to communicate effectively, taking the context and specific requirements for a target audience into consideration. These aspects will be discussed also in the scientific context, as well as business, and special cases, such as legal context - particularly with view to argumentation theory.

Communication is a truly transdisciplinary concept that involves knowledge from diverse fields such as biology, psychology, neuroscience, linguistics, sociology, philosophy, communication and information science. Students will learn what these different disciplines contribute to an understanding of communication and how theories from these fields can be applied in the real world. In the context of scientific communication, there will also be a focus on visual communication of data in different disciplines. Good practice examples will be contrasted with typical errors to facilitate successful communication also with view to the Bachelor's thesis.

Recommended Knowledge

- Ability and openness to engage in interactions

- Media literacy, critical thinking and a proficient handling of data sources

- Own research in academic literature

Intended Learning Outcomes

No	Competence	ILO
1	Analyze	Analyze communication processes in formal and informal contexts.
2	Identify	Identify challenges and failures in communication.
3	Design	Design communications to achieve specified goals to specific target groups.
4	Understand	Understand the principles of argumentation theory.
5	Use	Use data visualization in scientific communications.

Indicative Literature

- Joseph A. DeVito: The Interpersonal Communication Book (Global edition, 16th edition), 2022.
- Steven L. Franconeri, Lace M. Padilla, Priti Shah, Jeffrey M. Zacks, and Jessica Hullman: The Science of Visual Data Communication: What Works Psychological Science in the Public Interest, 22(3), 110–161, 2022.
- Douglas Walton: Argumentation Theory – A Very Short Introduction. In: Simari, G., Rahwan, I. (eds) Argumentation in Artificial Intelligence. Springer, Boston, MA, 2009.

Entry Requirements

Prerequisites	Logic (perspective I) Logic (perspective II) Causation and Correlation (perspective I) Causation and Correlation (perspective II)
Co-requisites	None
Additional Remarks	

Assessment and Completion

Components	Examination Type	Duration/ Length	Weight(%)	Minimum	ILOs
Argumentation, Data Visualization and Communication (perspective II)	Presentation	Digital submission (Asynchronous)	100	45%	1-5

Module Achievements: Asynchronous presentation on a topic relating to the major of the student, including a reflection including concept outlining the rationale for how arguments are selected and presented based on a particular target group for a particular purpose. The presentation shall be multimedial and include the presentation of data. The module achievement ensures sufficient knowledge about key concepts of effective communication including a reflection on the presentation itself.

8.2.7 Agency, Leadership, and Accountability

Module Name	**Agency, Leadership, and Accountability**
Module Code	2025-CTNS-NSK-09
Module ECTS	5
Study Semester	Mandatory status for: - 2025-S-ACS-BSc 5 Mandatory Elective status for: - 2025-F-ACS-BSc 6
Duration	1 Semester
Program Affiliation	2025-CT ()
Module Coordinator(s)	Prof. Dr. Jules Coleman

Forms of Learning and Teaching	
Independent Study	90
Online Lecture	35
Workload Hours	125 hours

Module Components	**Number**	**Type**	**CP**
Agency, Leadership, and Accountability	CTNS-09	Lecture (Online)	5

Module Description

Each of us is judged by the actions we undertake and held to account for the consequences of them. Sometimes we may be lucky and our bad acts don't have harmful effects on others. Other times we may be unlucky and reasonable decisions can lead to unexpected or unforeseen adverse consequences for others. We are therefore held accountable both for choices and for outcomes. In either case, accountability expresses the judgment that we bear responsibility for what we do and what happens as a result. But our responsibility and our accountability in these cases is closely connected to the idea that we have agency.

Agency presumes that we are the source of the choices we make and the actions that result from those choices. For some, this may entail the idea that we have free will. But there is scientific world view that holds that all actions are determined by the causes that explain them, which is the idea that if we knew the causes of your decisions in advance, we would know the decision you would make even before you made it. If that is so, how can your choice be free? And if it is not free, how can you be responsible for it? And if you cannot be responsible, how can we justifiably hold you to account for it?

These questions express the centuries old questions about the relationship between free will and a determinist world view: for some, the conflict between a scientific world view and a moral world view.

But we do not always act as individuals. In society we organize ourselves into groups: e.g. tightly organized social groups, loosely organized market economies, political societies, companies, and more. These groups have structure. Some individuals are given the responsibility of leading the group and of exercising authority. But one can exercise authority over others in a group merely by giving orders and threatening punishment for non-compliance.

Exercising authority is not the same thing as being a leader? For one can lead by example or by encouraging others to exercise personal judgment and authority. What then is the essence of leadership?

The module has several educational goals. The first is for students to understand the difference between actions that we undertake for which we can reasonably held accountable and things that we do but which we are not responsible for. For example, a twitch is an example of the latter, but so too may be a car accident we cause as a result of a heart attack we had no way of anticipating or controlling. This suggests the importance of control to responsibility. At the heart of personal agency is the idea of control. The second goal is for students to understand what having control means. Some think that the scientific view is that the world is deterministic, and if it is then we cannot have any personal control over what happens, including what we do. Others think that the quantum scientific view entails a degree of indeterminacy and that free will and control are possible, but only in the sense of being unpredictable or random. But then random outcomes are not ones we control either. So, we will devote most attention to trying to understand the relationships between control, causation and predictability.

But we do not only exercise agency in isolation. Sometimes we act as part of groups and organizations. The law often recognizes ways in which groups and organizations can have rights, but is there a way in which we can understand how groups have responsibility for outcomes that they should be accountable for. We need to figure out then whether there is a notion of group agency that does not simply boil down to the sum of individual actions. We will explore the ways in which individual actions lead to collective agency.

Finally we will explore the ways in which occupying a leadership role can make one accountable for the actions of others over which one has authority.

Intended Learning Outcomes

No	Competence	ILO
1	Understand	Understand and reflect how the social and moral world views that rely on agency and responsibility are compatible, if they are, with current scientific world views.
2	Understand	Understand how science is an economic sector, populated by large powerful organizations that set norms, fund research agendas.
3	Identify	Identify the difference between being a leader of others or of a group - whether a research group or a lab or a company - and being in charge of the group.
4	Learn	Learn to be a leader of others and groups. Understand that when one graduates one will enter not just a field of work but a heavily structured set of institutions and that one's agency and responsibility for what happens, what work gets done, its quality and value, will be affected accordingly.

Indicative Literature

- Hull, David L. "Science as a Process." Science as a Process. University of Chicago Press, 2010.
- Feinberg, Joel. "Doing & deserving; essays in the theory of responsibility." (1970).

Entry Requirements

Prerequisites	None
Co-requisites	None
Additional Remarks	None

Assessment and Completion

Components	**Examination Type**	**Duration/ Length**	**Weight(%)**	**Minimum**	**ILOs**
Agency, Leadership, and Accountability	Written Examination	120 minutes	100	45%	1-4

Module Achievements: None

9 Appendix

9.1 Intended Learning Outcomes Assessment Matrix

Applied Computer Science (BSc.)					Introduction to Computer Science	Programming in C and C++	Introduction to Data Science	Elements of Linear Algebra	Algorithms and Data Structures	Introduction to Cyber Physical Systems	Software Design and Prototyping	Elements of Calculus	Distributed Development	Databases and Web Services	Operating Systems	Data Analytics and Modeling	Probability and Random Processes	Software Engineering	Artificial Intelligence	Machine Learning	Internship/Start-Up	ACS Specialization Modules	Collaborative Software Project	Management Electives	Project	CT New Skills	Bachelor Thesis
Semester					1	1	1	1	2	2	2	2	1-2	3	3	3	3	4	4	4	5	5	5	4	5	3-6	4
Mandatory/ optional					m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	me	m	me	m	m	m
Credits					7.5	7.5	7.5	5	7.5	7.5	7.5	5	5	7.5	7.5	7.5	5	7.5	7.5	7.5	15	15	5	10	5	15	30
	Competencies*																										
Program Learning Outcomes	A	E	P	S																							
acquire Applied Computer Science knowledge in an independent, self-governed way	x	x	x		x	x		x	x	x	x			x	x			x	x	x	x	x	x		x		x
Work in teams distributed around the globe to analyze complex problems, to evaluate them, and to derive solutions	x	x			x	x	x	x	x	x	x		x	x	x	x		x	x	x		x			x	x	x
Comprehend the processes and tools of Software Engineering for collaborative, remote software and systems development	x	x				x					x		x			x		x					x		x		
Program software in C/C++ and understand algorithms;	x	x				x			x	x								x	x						x		x
Be able to use libraries and to generate software in core Computer Science areas	x	x	x						x	x				x	x	x			x	x					x		x
Apply suited mathematical methods	x	x	x				x	x		x		x				x	x			x					x		x
Understand operating systems, databases, and web services	x	x	x							x				x	x												
Comprehend methods from Artificial Intelligence and Machine Learning	x	x	x	x			x									x			x	x							
Understand the relation between software and its links to the physical world	x	x					x			x									x								
analyze data and to extract insights from it	x	x	x	x			x									x				x			x				
apply the acquired Software Engineering skills and Computer Science knowledge in collaborative, remote projects	x	x	x	x		x	x		x	x	x		x	x	x	x		x	x	x		x	x		x		
Use academic or scientific methods as appropriate in the field of Applied Computer Science such as defining research questions, justifying methods, collecting, assessing and interpreting relevant information, and drawing scientifically-founded conclusions that consider social, scientific and ethical insights;	x	x	x	x			x							x	x	x			x	x	x	x	x		x		x
Develop and advance solutions to problems and arguments in their subject area and defend these in discussions with specialists and non-specialists;		x	x	x						x			x					x			x	x	x	x	x	x	x
Engage ethically with academic, professional and wider communities and to actively contribute to a sustainable future, reflecting and respecting different views;		x	x	x			x						x			x										x	x
Take responsibility for their own learning, personal and professional development and role in society, evaluating critical feedback and self-analysis;		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Apply their knowledge and understanding to a professional context;	x	x	x								x		x								x		x	x	x		x
Take on responsibility in a diverse team;		x	x	x							x		x					x							x		
Adhere to and defend ethical, scientific and professional standards.		x	x	x																	x		x		x	x	x
Assessment Type																											
Written examination					x	x	x	x	x	x	x	x		x	x	x	x	x	x	x		x				x	
Term paper																								x			
Essay																											
Project report																					x		x		x		
Poster presentation																											
Laboratory report																											
Program Code																											
Oral examination																											
Presentation																								x		x	x
Practical assignment						x				x			x							x							
Project assessment											x			x				x	x			x					
Portfolio Assessments																											
Bachelor Thesis																											x
Module Achievements							x								x	x											

*Competencies: A-scientific/academic proficiency; E-competence for qualified employment; P-development of personality; S-competence for engagement in society