CONSTRUCTOR
UNIVERSITY

Study Program Handbook

Robotics and Intelligent Systems

Bachelor of Science

Subject-specific Examination Regulations for Robotics and Intelligent Systems (Fachspezifische Prüfungsordnung)

The subject-specific examination regulations for Robotics and Intelligent Systems 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 6).

Upon graduation, students in this program will receive a Bachelor of Science (BSc) degree with a scope of 180 ECTS (for specifics see Chapter 4 of this handbook).

Version	Valid as of	Decision	Details
Fall 2023- V1.3		July 24, 2025	Nonsubstantial minor change (RIS-2025-7): adding a new specialization module "Human-Robot Interaction"
Fall 2024- V1.2		Jan 22, 2025	Reversion of changes to specialization module area as they do not apply to the program.
Fall 2024- V1.1		Aug 29, 2024	Editorial changes
Fall 2024 – V1	Sep 01, 2024		
		Apr 26, 2023	Substantial change approved by the Academic Senate
		Jun 26, 2019	Originally approved by Academic Senate

Contents

1 Program Overview

1.1 Concept

1.1.1 The 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 and the minor option provide breadth while the university-wide general foundation and methods modules, optional German language and Humanities 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

Robotics and intelligent systems are more and more present in everyday life. Artificial intelligence and Machine learning are at the forefront of today's interconnected society.

Automation with some sort of embedded intelligence is now the norm rather than the exception. This program covers engineering methods and technologies that are relevant for freeing artificial mobile systems from permanent human supervision, to enable systems to perform autonomous intelligent operations. Application areas include the automotive and transport industries, robotics and automation, communication technologies, marine technology, and logistics. Hands-on experience with technical systems and methods is provided in first-class labs across the entire program.

During the first year, the foundations of the program are laid out, with programming courses, algorithms, and a comprehensive introduction to robotics and intelligent systems. The second year represents the core of the educational offering of the program, with courses focused on Robotics Systems (Robotics, Machine Learning), Automation and Control (Automation, Embedded Systems, Control Systems), and Intelligent Systems (Computer Vision, Artificial Intelligence). The RIS Lab and RIS project will complement the theoretical education, with use of both robotics simulators and real systems. During the third year, based on their specific interests and career goals, students can choose a variety of specialization courses to complement the core education in depth or breadth. Because robotics science is rooted in mathematics, students will take math methods modules covering calculus, linear algebra, probability theory, and numerical methods or discrete mathematics.

The job market for roboticists and experts in intelligent systems is increasing continuously, and all indications point to the growth of the sector in the near future. Because of the rapid changes in the field, it is important to focus the education on fundamental principles and in subfields of promising future relevance. Cross-disciplinary breadth and flexibility, as well as social and work organization skills are increasingly important. The minor option allows the combination of the education in robotics and intelligent systems with a different discipline, facilitating a cross- disciplinary specialization. The academic qualifications and personal profiles for academic and industrial careers differ. Constructor University's Robotics and Intelligent Systems program responds to the needs in both areas by offering a core Robotics and Intelligent Systems track designed for students who plan to join the industry, work in / found a start-up, or join graduate programs. A minor track allows students to obtain basic skills in specific application domains, which makes them well suited to work in specific industrial sectors.

1.2 Specific Advantages of Robotics and Intelligent Systems at Constructor University

- Robotics and Intelligent Systems is positioned in the School of Computer Science & Engineering. It has been designed with an interdisciplinary approach, incorporating concepts from various engineering disciplines such as Computer Science, Electrical Engineering, Mechanical Engineering, and Logistics.
- Although programs on Automation, Robotics, and Mechatronics exist in other universities, what makes Robotics and Intelligent Systems stand out is that, in addition to covering the aforementioned areas, it puts a special emphasis on the key concepts of Intelligence and Autonomy, which are important for the man-made systems of the future. Hence, students are given a solid background in fields such as Control Systems, Machine Learning, and Computer Vision.
- The Robotics and Intelligent Systems program is geared toward the world-renowned automation and robotics industry in Germany. As confirmed by keyword-searches on popular job-portals, engineers with additional skills in Vision, Machine Learning, and Robotics are much sought after by the well-established German and European automobile industry. A mandatory internship during the summer before the fifth

semester allows students to gain industrial experience and make contacts for potential future job opportunities.

- Cooperation with universities abroad allows ample choice for students interested in studying a semester abroad.
- The Robotics@Constructor initiative is a unique program to bring undergraduate students close to robotics systems, working with a variety of platforms. State-of-the-art, high-end equipment includes systems working in land, aerial, and marine domains, ranging from underwater robots to autonomous driving, and from humanoids to drones
- Based on their performance and interest, students can team up and participate in robotics competitions, e.g., the European Robotics League, receiving support and guidance from faculty members.
- Many faculty members have research groups that are well-funded by European Union (EU) and German Research Foundation (DFG) projects. Hence, ample opportunities exist for students to get involved and gain research experience.

1.3 Program-Specific Educational Aims

1.3.1 Qualification Aims

The main subject-specific qualification aim is to enable students to take up qualified employment in modern industries involving robotics, autonomous systems, machine learning, artificial intelligence, or to enter related graduate programs. Graduates of the Robotics and Intelligent Systems program have obtained the following competencies:

- Robotics and Intelligent Systems competence

Graduates are able to design and develop autonomous systems in a given application scenario, addressing both electrical engineering and computer science aspects. They can analyze, structure, and properly address complex problems. Graduates have the ability to construct and maintain complex robotics systems using a structured, analytic, and creative approach.

- Communication competence

Graduates are able to communicate subject-specific topics convincingly in both spoken and written form to fellow roboticists, experts in intelligent systems, industrial or academic colleagues, as well as to current and potential customers.

- Teamwork and project management competence

Graduates are able to work effectively in a team and to organize workflows in complex development efforts. They are familiar with tools that support the development, testing, and maintenance of complex intelligent systems and they can take design decisions in a constructive way.

- Learning competence

Graduates have acquired a solid foundation enabling them to learn effectively and to stay up to date with the latest developments in the fast-changing field of robotics and intelligent systems.

- Personal and professional competence

Graduates are able to develop a professional profile, justify professional decisions on the basis of theoretical and methodical knowledge, and critically on reflect their behavior, also with respect to its consequences for society.

During the design of the program, national guidelines published by the Gesellschaft für Informatik (GI) (GI: Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen, July 2016) and international guidelines published jointly by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) (ACM/IEEE: Computer Science Curricula 2013, December 2013) have been consulted.

1.3.2 Intended Learning Outcomes

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

1. design basic electronics circuits
2. think in an analytic way at multiple levels of abstraction
3. develop, analyze and implement algorithms using modern software engineering methods.
4. demonstrate knowledge of kinematics and dynamics of multi-body systems
5. design and develop linear and nonlinear control systems
6. design basic electronics circuits
7. examine physical problems, apply mathematical skills to find possible solutions and assess them critically
8. show competence about operational principles of motors and drives
9. design and develop machine learning algorithms and techniques for pattern-recognition, classification, and decision-making under uncertainty;
10. design and develop computer vision algorithms for inferring 3D information from camera images, and for object recognition and localization
11. model common mechanical and electrical systems that are part of intelligent mobile systems
12. design robotics systems and program them using popular robotics software frameworks
13. use academic or scientific methods as appropriate in the field of Robotics and Intelligent Systems 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
14. develop and advance solutions to problems and arguments in their subject area and defend these in discussions with specialists and non-specialists;
15. engage ethically with the academic, professional, and wider communities and to actively contribute to a sustainable future, reflecting and respecting different views;
16. take responsibility for their own learning, personal, and professional development and role in society, evaluating critical feedback and self-analysis;
17. apply their knowledge and understanding to a professional context;
18. work effectively in a diverse team and take responsibility in a team;
19. adhere to and defend ethical, scientific, and professional standards.

1.4 Career Options and Support

Career options include areas such as research and development or management tracks in the automotive and transport, robotics and automation, communication technologies, marine technology and logistics industries. Given the increasing need for automation of daily life tasks through intelligent mobile systems, there is a significant number of career options in addition to the core options that are covered in the program.

The Robotics and Intelligent Systems program matches scientific content with real-world use cases. This is a strength of Constructor University's offering, to introduce students to real- world applications.

Field trips to and participation in robotics competitions significantly contribute to bringing students closer to the market and to real challenges, in addition to being an excellent opportunity for professional networking.

Companies which hired recent graduates of the IMS program (Intelligent Mobile Systems, the former name of RIS) include Cambio CarSharing Deutschland, Daimler AG, Klöckner Desma GmbH, Objective Software GmbH, and Ubimax.

Several graduate programs have offered a position to IMS students, including the Master in Artificial Intelligence, offered by Universita' della Svizzera Italiana (Switzerland), the Erasmus Mundus Joint Master Degree on Advanced Robotics, offered by Centrale Nantes (France), University of Genoa (Italy), Warsaw University of Technology (Poland), and Jaume I University (Spain), as well as the Master in Robotics, offered by Heriot-Watt University (Scotland, UK).

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. Francesco Maurelli

Professor of Marine Systems and Robotics Email:

fmaurelli@constructor.university

or visit our website: https://constructor.university/programs/undergraduate-education/robotics-intelligent-systems

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. The unique CONSTRUCTOR Track, offered across all undergraduate study programs, provides comprehensive tailor-made modules designed to achieve and foster career competency. Additionally, a mandatory internship of at least two months after the second year of study and the possibility to study abroad for one semester give students the opportunity 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 an extensive and flexible choice of study plans to meet the educational aims of their major as well as minor study interests and 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).

2.2 The Constructor University 4C Model

Constructor University offers study programs that comply with the regulations of the European Higher Education Area. All study programs are structured according to the European Credit Transfer System (ECTS), which facilitates credit transfer between academic institutions. The three-year undergraduate programs involve six semesters of study with a total of 180 ECTS credit points (CP). The undergraduate curricular structure follows an innovative and student- centered modularization scheme, the 4C Model. It groups the disciplinary content of the study program in three overarching themes, CHOICE-CORE-CAREER according to the year of study, while the university-wide CONSTRUCTOR Track is dedicated to multidisciplinary content dedicated to methods as well as intellectual skills and is integrated across all three years of study. The default module size is 5 CP, with smaller 2.5 CP modules being possible as justified exceptions, e.g., if the learning goals are more suitable for 2.5 CP and the overall student workload is balanced.

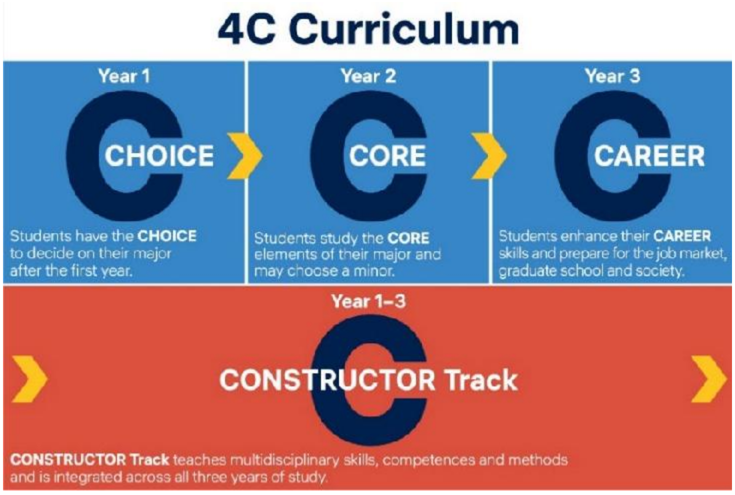

4C Curriculum
Year 1
CHOICE
Students have the CHOICE to decide on their major after the first year.
Year 2
CORE
Students study the CORE elements of their major and may choose a minor.
Year 3
CAREER
Students enhance their CAREER skills and prepare for the job market, graduate school and society.
Year 1–3
CONSTRUCTOR Track
CONSTRUCTOR Track teaches multidisciplinary skills, competences and methods and is integrated across all three years of study.

Figure 1: The Constructor University 4C-Model

2.2.1 Year 1 – CHOICE

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 select introductory modules for a total of 45 CP from the CHOICE area of a variety of study programs, of which 15-45 CP will belong to their intended major. A unique feature of our curricular structure allows students to select their major freely upon entering Constructor University. 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 in deciding on their major study program.

To pursue Robotics and Intelligent Systems as a major, the following CHOICE modules (45 CP) need to be taken:

- CHOICE Module: Mathematical and Physical Foundations of Robotics I (m, 7.5 CP)
- CHOICE Module: Mathematical and Physical Foundations of Robotics II (m, 7.5 CP)
- CHOICE Module: Programming in C and C++ (m, 7.5 CP)
- CHOICE Module: Algorithms and Data Structures (m, 7.5 CP)
- CHOICE Module: General Electrical Engineering (m, 7.5 CP)
- CHOICE Module: Digital Systems and Computer Architecture (m, 7.5 CP)

Students can still change to another major at the beginning of their second year of studies, provided they have taken the corresponding mandatory CHOICE modules in their first year of studies. All students must participate in an entry advising session with their Academic Advisors to learn about their major change options and consult their Academic Advisor during the first year of studies prior to changing their major.

To allow further major changes after the first semester the students are strongly recommended to register for the CHOICE modules of one of the following study programs:

- Electrical and Computer Engineering (ECE):
  CHOICE Module: General Electrical Engineering I (m, 7.5 CP)
  CHOICE Module: General Electrical Engineering II (m, 7.5 CP)
  CHOICE Module: Classical Physics (m, 7.5 CP)
  CHOICE Module: Digital Systems and Computer Architecture (m, 7.5 CP)
  CHOICE Module: Foundations of Communications and Electronics (m, 7.5 CP)

2.2.2 Year 2 – CORE

In their second year, students take a total of 45 CP from a selection of in-depth, discipline- specific CORE modules. Building on the introductory CHOICE modules and applying the methods and skills acquired so far (see 2.3.1), 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.

To pursue Robotics and Intelligent Systems as a major, 30 CP from the following mandatory (m) and mandatory elective (me) CORE modules need to be taken:

- CORE Module: Robotics (m, 5 CP)
- CORE Module: Artificial Intelligence (m, 5 CP)
- CORE Module: RIS Lab (me, 5 CP)
- CORE Module: Automation (me, 5 CP)
- CORE Module: Embedded Systems (me, 5 CP)

- CORE Module: Control Systems (me, 5 CP)
- CORE Module: Computer Vision (me, 5 CP)
- CORE Module: Machine Learning (m, 5 CP)
- CORE Module: RIS Project (m, 5 CP)

15 CP can be selected according to interest and/or with the aim of pursuing a minor in Computer Science, or students complement their studies by taking all of the above listed mandatory elective CORE modules.

Robotics and Intelligent Systems students can take CORE modules (or more advanced Specialization modules) from other programs, which allows them to incorporate a minor study track into their undergraduate education, within the 180 CP required for a bachelor's degree. The educational aims of a minor are to broaden the students' knowledge and skills, support the critical reflection of statements in complex contexts, foster an interdisciplinary approach to problem-solving, and to develop an individual academic and professional profile in line with students' strengths and interests. This extra qualification will be highlighted in the transcript.

The Academic Advising Coordinator, Academic Advisor, and the Study Program Chair of the minor study program support students in the realization of their minor selection; the consultation with the Academic Advisor is mandatory when choosing a minor.

According to the default study plan RIS students have the option to pursue a minor in Computer Science.

This requires Robotics and Intelligent Systems students to

- substitute the three mandatory elective Robotics and Intelligent Systems CORE modules (15 CP) in the second year with the default minor CORE modules of the Computer Science program.

The requirements for the specific minors are described in the handbook of the study program offering the minor (Chapter 3.2) and are marked in the respective Study and Examination Plans.

2.2.3 Year 3 – CAREER

During their third year, students prepare and make decisions about their career path after graduation. To explore available choices and to gain professional experience, students undertake a mandatory summer internship. The third year of studies allows RIS students to take Specialization modules within their discipline, but also focuses on the responsibility of students beyond their discipline (see CONSTRUCTOR Track).

The fifth semester also opens a mobility window for a diverse range of study abroad options. Finally, the sixth semester is dedicated to fostering the students' research experience by involving them in an extended Bachelor thesis project.

The third year of studies allows RIS students to take Specialization modules within their discipline, but also focuses on the responsibility of students beyond their discipline (see CONSTRUCTOR Track).

2.2.3.1 Internship / Start-up 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 role in employment and society, and find 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 of their business plans.

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

2.2.3.2 Specialization Modules

In the third year of their studies, students take 15 CP from major-specific or major-related, advanced 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 which students can make free selections during their fifth and sixth semester. The default Specialization Module size is 5 CP, with smaller 2.5 CP modules being possible as justified exceptions.

To pursue RIS as a major, 15 CP from the following mandatory elective Specialization Modules need to be taken:

- RIS Specialization: Marine Robotics (me, 5 CP)
- RIS Specialization: Optimization (me, 5 CP)
- RIS Specialization: Human-Robot Interaction (me, 5 CP)
- CS Specialization: Distributed Algorithms (me, 5 CP)
- CS Specialization: Computer Graphics (me, 5 CP)
- CS Specialization: Web Application Development (me, 5 CP)
- CS CORE: Software Engineering (me, 7.5 CP)
- CS CORE: Databases (me, 7.5 CP)
- ECE Specialization: Digital Design (me, 5 CP)
- ECE CORE: PCB design and measurement automation (me, 5 CP)
- ECE CORE: Information Theory (me, 5 CP)
- MATH Specialization: Stochastic Processes (me, 5 CP)
- MATH Specialization: Stochastic Modeling and Financial Mathematics (me, 5 CP)
- IEM CORE: Operations Research (me, 5 CP)
- DE ELECTIVE: Parallel and Distributed Computing (me, 5 CP)
- ACS Specialization Module: Human Computer Interaction (me, 5 CP)

Available for RIS students minoring in the respective study program that meet the pre-requisites / co-requisites[1]

- CS Specialization: Image Processing (me, 5 CP)
- CS Specialization: Automata, Computability, and Complexity (me, 7.5 CP)
- CS Specialization: Computer Networks (me, 5 CP)
- CS Specialization: Operating Systems (me, 7.5 CP)

[1] For module descriptions, see the respective handbook offering the modules.

- ECE Specialization: Electronics (me, 5 CP)
- ECE Specialization: Digital Signal Processing (me, 7.5 CP)
- ECE Specialization: Signals and Systems (me, 7.5 CP)
- IEM Specialization: Industry 4.0 and Blockchain Technologies (me, 5 CP)
- IEM Specialization: Process Modeling and Simulation (me, 5 CP)

In case of students pursuing a minor, the CORE modules of the Robotics and Intelligent Systems program which are substituted for the minor modules are also eligible Specialization Modules.

2.2.3.3 Study Abroad

Students have the opportunity to study abroad for a semester to extend their knowledge and abilities, broaden their horizons and reflect on their values and behavior in a different context as well as on their role in a global society. For a semester abroad (usually the 5th semester), modules related to the major with a workload equivalent to 22.5 CP must be completed. Modules recognized as study abroad CP need to be pre-approved according to Constructor University study abroad procedures. Several exchange programs allow students to directly enroll at prestigious partner institutions worldwide. Constructor University's participation in Erasmus+, the European Union's exchange program, provides an exchange semester at a number of European universities that include Erasmus study abroad funding.

For further information, please contact the International Office (https://constructor.university/student-life/study-abroad/international-office).

RIS students that wish to pursue a study abroad in their 5th semester are required to select their modules at the study abroad partners such that they can be used to substitute between 10-15 CP of major-specific Specialization modules and between 5-15 CP of modules equivalent to the non-disciplinary New Skills modules (see CONSTRUCTOR Track). In their 6th semester, according to the study plan, returning study-abroad students complete the Bachelor Thesis/Seminar module (see next section), they take any missing Specialization modules to reach the required 15 CP in this area, and they take any missing New Skills modules to reach 15 CP in this area.

2.2.3.4 Bachelor Thesis/Seminar Module

This module is a mandatory graduation requirement for all undergraduate students. It consists of two module components in the major study program guided by a Constructor faculty member: the Bachelor Thesis (12 CP) and a Seminar (3 CP). 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. In the seminar, students present and discuss their theses in a course environment and reflect on their theoretical or experimental approach and conduct. They learn to present their chosen research topics concisely and comprehensively in front of an audience and to explain their methods, solutions, and results to both specialists and non-specialists.

2.3 The CONSTRUCTOR Track

The CONSTRUCTOR Track is another important feature of Constructor University's educational model. The Constructor Track runs orthogonal to the disciplinary CHOICE, CORE, and CAREER modules across all study years and is an integral part of all undergraduate study programs. It provides an intellectual tool kit for lifelong learning and encourages the use of diverse methodologies to approach cross-disciplinary problems. The CONSTRUCTOR track contains Methods, New Skills and German Language and Humanities modules.

2.3.1 Methods Modules

Methods such as mathematics, statistics, programming, data handling, presentation skills, academic writing, and scientific and experimental skills are offered to all students as part of the Methods area in their curriculum. The modules that are specifically assigned to each study programs equip students with transferable academic skills. They convey and practice specific methods that are indispensable for each students' chosen study program. Students are required to take 20 CP in the Methods area. The size of all Methods modules is 5 CP.

To pursue Robotics and Intelligent Systems as a major, the following Methods modules (20 CP) are mandatory

- Methods Module: Elements of Linear Algebra (me, 5 CP)
- Methods Module: Elements of Calculus (me, 5 CP)
- Methods Module: Probability and Random Processes (m, 5 CP)

Students who have a strong mathematical background can also choose Matrix Algebra and Advanced Calculus I and II (me, 5 CP each) instead of Elements of Linear Algebra and Elements of Calculus.

For the remaining 5 CP RIS students can choose between the Methods modules

- Methods Module: Numerical Methods (me, 5 CP)

and the Mathematics CORE module:

- CORE Module: Discrete Mathematics (me, 5 CP)

2.3.2 New Skills Modules

This part of the curriculum constitutes an intellectual and conceptual tool kit that cultivates the capacity for a particular set of intellectual dispositions including curiosity, imagination, critical thought, and transferability. It nurtures a range of individual and societal capacities, such as self-reflection, argumentation and communication. Finally, it introduces 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:

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

These modules will be offered with two different perspectives of which the students can choose. The module perspectives are independent modules which examine the topic from different point of views. Please see the module description for more details.

In the third year, students take three 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 module is mandatory for all students:

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

This module will also be offered with two different perspectives of which the students can choose.

In their fifth semester, students may choose between:

- New Skills Module: Linear Model/Matrices (me, 5 CP) and
- New Skills Module: Complex Problem Solving (me, 5 CP).

The sixth semester also contains the choice between two modules, namely:

- New Skills Module: Agency, Leadership and Accountability (me, 5 CP) and
- New Skills Module: Community Impact Project (me, 5 CP).

Students who study abroad during the fifth semester and are not substituting the mandatory Argumentation, Data Visualization and Communication module, are required to take this module during their sixth semester. Students who remain on campus are free to take the Argumentation, Data Visualization and Communication module in person in either the fifth or sixth.

2.3.3 German Language and Humanities Modules

German language abilities foster students' intercultural awareness and enhance their employability in their host country. They are also beneficial for securing mandatory internships (between the 2nd and 3rd year) in German companies and academic institutions. Constructor University supports its students in acquiring basic as well as advanced German skills in the first year of the Constructor Track. Non-native speakers of German are encouraged to take 2 German modules (2.5 CP each) but are not obliged to do so. Native speakers and other students not taking advantage of this offering take alternative modules in Humanities in each of the first two semesters:

- Humanities Module: Introduction to Philosophical Ethics (me, 2.5 CP)
- Humanities Module: Introduction to the Philosophy of Science (me, 2.5 CP)
- Humanities Module: Introduction to Visual Culture (me, 2.5 CP)

3 Robotics and Intelligent Systems as a Minor

3.1 Qualification Aims

Students obtaining a minor in Robotics and Intelligent Systems learn the basic principles of intelligent systems, including elements of both hardware and software. They obtain an understanding of how current robotics systems are designed and function. Upon completion of the minor, they will have obtained sufficient knowledge about robotics and intelligent systems concepts such that they can effectively work together with professional roboticists and experts in intelligent systems. Students obtaining a minor in Robotics and Intelligent Systems can help to drive and advise on the automation processes, which are at the forefront of industrial interest currently and are expected to remain so for the foreseeable future.

3.1.1 Intended Learning Outcomes

With a minor in Robotics and Intelligent Systems, students will be able to

1. develop solutions to problems in the automation, robotics, and intelligent systems domains in close collaboration with professionals;
2. design and develop software of moderate complexity for robotics and intelligent systems;
3. design and develop basic algorithms and techniques for pattern-recognition, classification, and decision-making under uncertainty.

3.2 Module Requirements

A minor in Robotics and Intelligent Systems requires 30 CP. The default option to obtain a minor in Robotics and Intelligent Systems is marked in the Study and Examination Plan. It includes the following mandatory CHOICE and CORE modules:

- CHOICE Module: Mathematical and Physical Foundations of Robotics I (m, 7.5 CP)
- CHOICE Module: Mathematical and Physical Foundations of Robotics II (m, 7.5 CP)
- CORE Module: Robotics (m, 5 CP)
- CORE Module: Artificial Intelligence (m, 5 CP)
- CORE Module: RIS Lab (m, 5 CP)

Upon consultation with the Academic Advisor and the RIS Study Program Chair, individual CORE modules from the default minor can be replaced by other advanced modules (CORE or Specialization) from the RIS major.

3.3 Degree

After successful completion, the minor in Robotics and Intelligent Systems will be listed on the final transcript under PROGRAM OF STUDY and BA/BSc – [name of the major] as "(Minor: Robotics and Intelligent Systems)."

4 Robotics and Intelligent Systems Undergraduate Program Regulations

4.1 Scope of these Regulations

The regulations in this handbook are valid for all students who entered the Robotics and Intelligent Systems undergraduate program at Constructor University in Fall 2024. In case of a conflict between the regulations in this handbook and the general Policies for Bachelor Studies, the latter shall applies (https://constructor.university/student-life/student-services/university- 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).

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

4.2 Degree

Upon successful completion of the study program, students are awarded a Bachelor of Science degree in Robotics and Intelligent Systems.

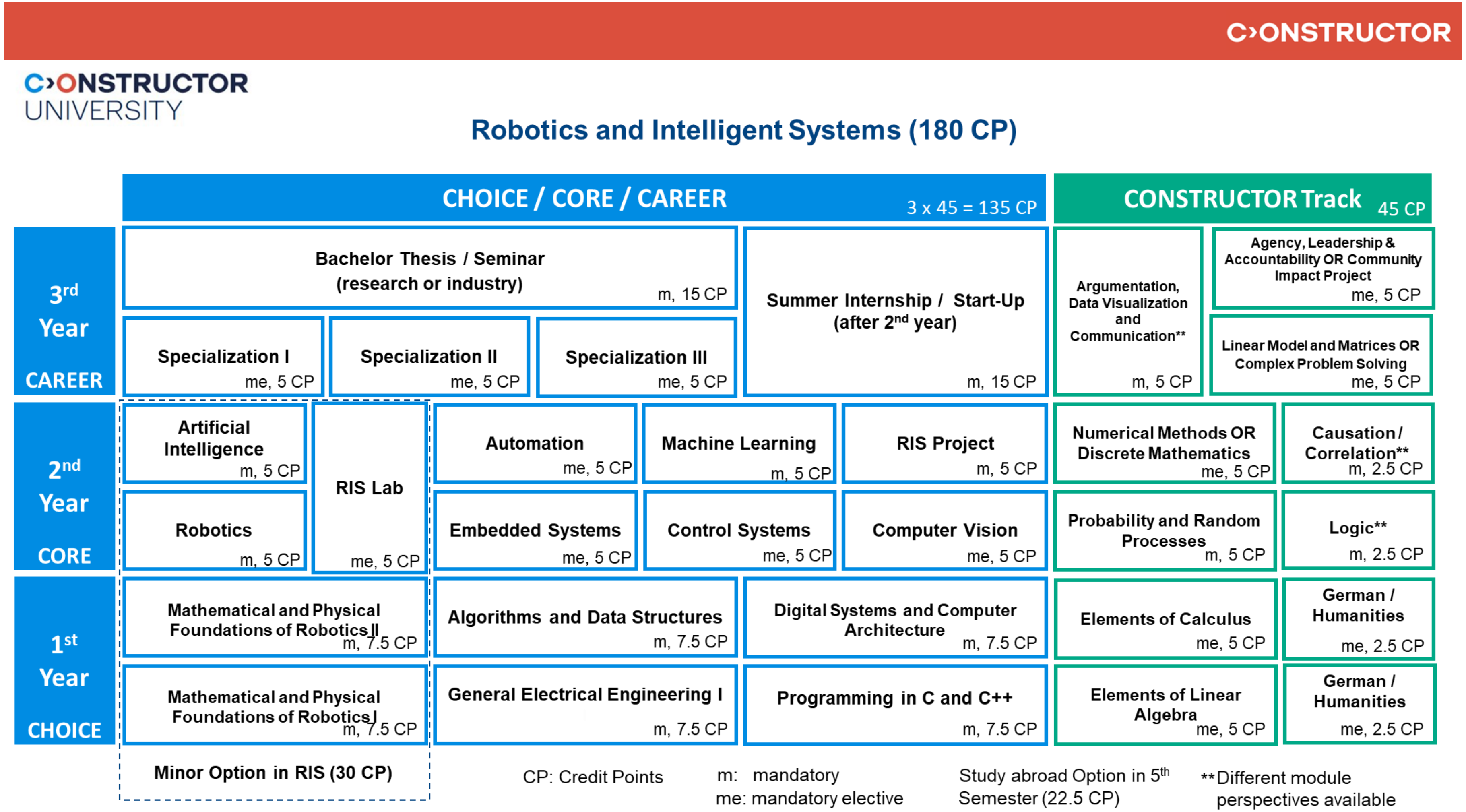
4.3 Graduation Requirements

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 6 of this handbook.

Schematic Study Plan for Robotics and Intelligent Systems

Figure 2 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

Robotics and Intelligent Systems (180 CP)

CHOICE / CORE / CAREER 3 x 45 = 135 CP

CONSTRUCTOR Track 45 CP

3rd Year CAREER

Bachelor Thesis / Seminar (research or industry) m, 15 CP

Specialization I me, 5 CP

Specialization II me, 5 CP

Specialization III me, 5 CP

Summer Internship / Start-Up (after 2nd year) m, 15 CP

Argumentation, Data Visualization and Communication** m, 5 CP

Agency, Leadership & Accountability OR Community Impact Project me, 5 CP

Linear Model and Matrices OR Complex Problem Solving me, 5 CP

2nd Year CORE

Artificial Intelligence m, 5 CP

Robotics m, 5 CP

RIS Lab me, 5 CP

Automation me, 5 CP

Machine Learning m, 5 CP

RIS Project m, 5 CP

Embedded Systems me, 5 CP

Control Systems me, 5 CP

Computer Vision me, 5 CP

Numerical Methods OR Discrete Mathematics me, 5 CP

Causation / Correlation** m, 2.5 CP

Probability and Random Processes m, 5 CP

Logic** m, 2.5 CP

1st Year CHOICE

Mathematical and Physical Foundations of Robotics II m, 7.5 CP

Mathematical and Physical Foundations of Robotics I m, 7.5 CP

Algorithms and Data Structures m, 7.5 CP

General Electrical Engineering I m, 7.5 CP

Digital Systems and Computer Architecture m, 7.5 CP

Programming in C and C++ m, 7.5 CP

Elements of Calculus me, 5 CP

Elements of Linear Algebra me, 5 CP

German / Humanities me, 2.5 CP

German / Humanities me, 2.5 CP

Minor Option in RIS (30 CP)

CP: Credit Points

m: mandatory
me: mandatory elective

Study abroad Option in 5th Semester (22.5 CP)

**Different module perspectives available

Figure 2: Schematic Study Plan for RIS

6 Study and Exam Plan

Robotics and Intelligent Systems (RIS) BSc

Matriculation Fall 2024

	Program-Specific Modules	Type	Assessment	Period	Status[1]	Sem.	CP
Year 1 - CHOICE *Take the mandatory CHOICE modules listed below*							**45**
							15
CH-221	**Module: Mathematical and Physical Foundations of Robotics I (default minor)**				**m**	**1**	**7.5**
CH-221-A	Mathematical and Physical Foundations of Robotics I	Lecture	Written examination	Examination period			5
CH-221-B	Mathematical and Physical Foundations of Robotics I - Lab	Lab					2.5
CH-222	**Module: Mathematics and Physical Foundations of Robotics II (default minor)**				**m**	**2**	**7.5**
CH-222-A	Mathematical and Physical Foundations of Robotics II	Lecture	Written examination	Examination period			5
CH-222-B	Mathematical and Physical Foundations of Robotics II - Lab	Lab					2.5
CH-234	**Module: Digital Systems and Computer Architecture**				**m**	**2**	**7.5**
CH-234-A	Digital Systems and Computer Architecture	Lecture	Written examination	Examination period			5
CH-234-B	Digital Systems and Computer Architecture Tutorial	Tutorial					2.5
CH-210	**Module: General Electrical Engineering I**				**m**	**1**	**7.5**
CH-210-A	General Electrical Engineering I	Lecture	Written examination	Examination period			5
CH-210-B	General Electrical Engineering Lab I	Lab	Laboratory report	During the semester			2.5
CH-230	**Module: Programming in C and C++**				**m**	**1**	**7.5**
CH-230-A	Programming in C and C++	Lecture	Written examination	Examination period			5
CH-230-B	Programming in C and C++ Tutorial	Tutorial	Program code	During the semester			2.5
CH-231	**Module: Algorithms and Data Structures**				**m**	**2**	**7.5**
CH-231-A	Algorithms and Data Structures	Lecture	Written examination	Examination period			
Year 2 - CORE *Take all CORE modules listed below or replace mandatory elective ("me") modules with the default minor CORE modules of Computer Science.* [2]							**45**
	Unit: Robotics						**15**
CO-540	**Module: Robotics *(default minor)***				**m**	**3**	**5**
CO-540-A	Robotics	Lecture	Written examination	Examination period			
CO-541	**Module: Machine Learning**				**m**	**4**	**5**
CO-541-A	Machine Learning	Lecture	Written examination	Examination period			
CO-542	**Module: Robotics and Intelligent Systems Lab *(default minor)***				**me/m**[4]	**3-4**	**5**
CO-542-A	RIS lab I	Lecture/Lab	Laboratory Report	During the semester		3	2.5
CO-542-B	RIS Lab II	Lecture/Lab	Laboratory Report			4	2.5
	Unit: Automation and Control						**15**
CO-543	**Module: Automation**				**me**	**4**	**5**
CO-543-A	Automation	Lecture	Written examination	Examination period			
CO-544	**Module: Embedded Systems**				**me**	**3**	**5**
CO-544-A	Embedded Systems	Lecture/Lab	Project Assessment	During the semester			
CO-545	**Module: Control Systems**				**me**	**3**	**5**
CO-545-A	Control Systems	Lecture	Written examination	Examination period			
	Unit: Intelligent Systems						**15**
CO-546	**Module: Computer Vision**				**me**	**3**	**5**
CO-546-A	Computer Vision	Lecture/Lab	Written examination	Examination period			
CO-547	**Module: Artificial Intelligence *(default minor)***				**m**	**4**	**5**
CO-547-A	Artificial Intelligence	Lecture	Written examination	Examination period			
CO-548	**Module: RIS project**				**m**	**4**	**5**
CO-548-A	RIS Project	Project/Lab	Project Report / Presentation	During the semester			

	CONSTRUCTOR Track Modules (General Education)	Type	Assessment	Period	Status[1]	Sem.	CP
							15
	Unit: Methods						**10**
CTMS-MAT-24	**Module: Elements of Linear Algebra**				**me**	**1**	**5**
CTMS-24	Elements of Linear Algebra	Lecture	Written examination	Examination period			
CTMS-MAT-25	**Module: Elements of Calculus**				**me**	**2**	**5**
CTMS-25	Elements of Calculus	Lecture	Written examination	Examination period			
Students who have a strong mathematical background can also choose:							
CTMS-MAT-22	**Module: Matrix Algebra & Advanced Calculus I**				**me**	**1**	**5**
CTMS-22	Matrix Algebra & Advanced Calculus I	Lecture	Written examination	Examination period			
CTMS-MAT-23	**Module: Matrix Algebra & Advanced Calculus II**				**me**	**2**	**5**
CTMS-23	Matrix Algebra & Advanced Calculus II	Lecture	Written examination	Examination period			
	Unit: German Language and Humanities (choose one module for each sememster)						**5**
German is default language and open to Non-German speakers (on campus and online). [5]							
CTLA-xxx	**Module: Language 1**				**me**	**1**	**2,5**
CTLA-xxx	Language 1	Seminar	Various	Various			
CTLA-xxx	**Module: Language 2**				**me**	**2**	**2,5**
CTLA-xxx	Language 2	Seminar	Various	Various			
CTHU-HUM-001	**Humanities Module: Introduction to Philosophical Ethics**				**me**	**2**	**2,5**
CTHU-001	Introduction to Philosophical Ethics	Lecture (online)	Written examination	Examination period			
CTHU-HUM-002	**Humanities Module: Introduction to the Philosophy of Science**				**me**	**1**	**2,5**
CTHU-002	Introduction to the Philosophy of Science	Lecture (online)	Written examination	Examination period			
CTHU-HUM-003	**Humanities Module: Introduction to Visual Culture**				**me**	**2**	**2,5**
CTHU-003	Introduction to Visual Culture	Lecture (online)	Written examination	Examination period			
							15
	Unit: Methods						**10**
CTMS-MAT-12	**Module: Probability and Random Processes**				**m**	**3**	**5**
CTMS-12	Probability and Random Processes	Lecture	Written examination	Examination period			
Choose one of the following modules							
CTMS-MAT-13	**Module: Numerical Methods**				**me**	**4**	**5**
CTMS-13	Numerical Methods	Lecture	Written examination	Examination period			
CO-501-A	**Module: Discrete Mathematics**				**me**	**4**	**5**
CO-501-A	Discrete Mathematics	Lecture	Written examination	Examination period			
	Unit: New Skills						**5**
Choose one of the two modules							
CTNS-NSK-01	**Module: Logic (perspective I)**				**me**	**3**	**2,5**
CTNS-01	Logic (perspective I)	Lecture (online)	Written Examination	Examination period			
CTNS-NSK-02	**Module: Logic (perspective II)**				**me**		**2,5**
CTNS-02	Logic (perspective II)	Lecture (online)	Written Examination	Examination period			
Choose one of the two modules							
CTNS-NSK-03	**Module: Causation and Correlation (perspective I)**				**me**	**4**	**2,5**
CTNS-03	Causation and Correlation (perspective I)	Lecture (online)	Written Examination	Examination period			
CTNS-NSK-04	**Module: Causation and Correlation (perspective II)**				**me**	**4**	**2,5**
CTNS-04	Causation and Correlation (perspective II)	Lecture (online)	Written Examination	Examination period			

Year 3 - CAREER							**45**
CA-INT-900	**Module: Summer Internship**				**m**	**4/5**	**15**
CA-INT-900-0	Summer Internship	Internship	Report/Business Plan and Presentation	During the 5th Semester			
CA-RIS-800	**Module: Thesis / Seminar RIS**				**m**	**6**	**15**
CA-RIS-800-T	Thesis RIS	Thesis	Thesis and Presentation	15th of May			12
CA-RIS-800-S	Thesis Seminar RIS	Seminar		During the semester			3
	Unit: Specialization RIS				**m**	**5/6**	**15**
Take a total of 15 CP of specialization modules							
CA-S-RIS-801	Marine Robotics	Lecture/Lab	Oral examination	Examination period	me	6	5
CS-S-RIS-802	Human-Robot Interaction	Lecture	Project Assessment	Examination period	me	5	5
CS-S-RIS-803	Optimization	Lecture	Written examination	Examination period	me	6	5
CA-S-xxx	Specialization elective (from ACS, CS, ECE, MMDA, IEM, DE study programs)[3]	Various	Various	Various	me	5/6	5

							15
	Unit: New Skills						**10**
Choose one of the two modules							
CTNS-NSK-05	**Module: Linear Model and Matrices**				**me**	**5**	**5**
CTNS-05	Linear Model and Matrices	Seminar (online)	Written examination	Examination period			
CTNS-NSK-06	**Module: Complex Problem Solving**				**me**	**5**	**5**
CTNS-06	Complex Problem Solving	Lecture (online)	Written examination	Examination period			
Choose one of the two modules							
CTNS-NSK-07	**Module: Argumentation, Data Visualization and Communication (perspective I)**				**me**	**5/6**	**5**
CTNS-07	Argumentation, Data Visualization and Communication (perspective I)	Lecture (online)	Written examination	Examination period		5	
CTNS-NSK-08	**Module: Argumentation, Data Visualization and Communication (perspective II)**				**me**	**5/6**	**5**
CTNS-08	Argumentation, Data Visualization and Communication (perspective II)	Lecture (online)	Written examination	Examination period		6	
Choose one of the two modules							
CTNS-NSK-09	**Module: Agency, Leadership, and Accountability**				**me**	**6**	**5**
CTNS-09	Agency, Leadership, and Accountability	Lecture (online)	Written examination	Examination period			
CTNS-CIP-10	**Module: Community Impact Project**				**me**	**6**	**5**
CTNS-10	Community Impact Project	Project	Project	During the Sememster			
Total CP							**180**

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

[2] For a full listing of all CHOICE / CORE / CAREER / CONSTRUCTOR Track modules please consult the CampusNet online catalogue and /or the study program handbooks.

[3] For details please see the program handbook

[4] This module is me for students majoring in RIS, but m for students minoring in RIS

[5] German native speakers will have alternatives to the language courses (in the field of Humanities).

Figure 3: Study and Examination Plan

7 Robotics and Intelligent Systems Modules

7.1 Mathematical and Physical Foundations of Robotics I

Module Name **Mathematical and Physical Foundations of Robotics I**	**Module Code** CH-221	**Level (type)** Year 1 (CHOICE)	**CP** 7.5

Module Components			
Number	Name	Type	CP
CH-221-A	Mathematical and Physical Foundations of Robotics I	Lecture	5
CH-221-B	Mathematical and Physical Foundations of Robotics I - Lab	Lab	2.5

Module Coordinator Prof. Dr. Francesco Maurelli	**Program Affiliation** - Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory for RIS and minor RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall)	- Lecture (35 hours) - Lab (17.5 hours) - Private study (115 hours) - Exam preparation (20 hours)
☒ None	☒ None			
			Duration 1 semester	**Workload** 187.5 hours

Recommendations for Preparation Review basic mathematical concepts.

Content and Educational Aims This module represents an initial introduction to robotics and intelligent systems, starting from the basics of mathematics and physics applied to simple robotics scenarios. It will cover transformation matrices and quaternions for reference systems. The module will then cover kinematics of a point and of a rigid body. Students will then learn the basics of trajectory planning and robotic systems. The lab is designed to guide students through practical hands-on work with various components of intelligent systems. It will focus on the interfacing of a microcontroller with commonly used sensors and actuators.

Intended Learning Outcomes

By the end of this module, successful students will be able to

1. compute 3D transformations;
2. understand and apply quaternion operations;
3. understand and apply kinematics laws of a point;
4. understand and apply kinematics laws of a rigid body;
5. understand and apply trajectory planning techniques;
6. program the open-source electronic prototyping platform Arduino;
7. interface Arduino to several different sensors and actuators.

Indicative Literature

R. V. Roy, Advanced Engineering Dynamics. R. V. Roy, 2015.

R. N. Jazar, Theory of Applied Robotics. Springer, 2010.

Usability and Relationship to other Modules

- This module is the foundation of the CORE modules in the following years.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Module achievement: Lab report

Completion: To pass this module, the examination has to be passed with at least 45%.

7.2 Mathematical and Physical Foundations of Robotics II

Module Name	**Module Code**	**Level (type)**	**CP**
Mathematical and Physical Foundations of Robotics II	CH-222	Year 1 (CHOICE)	7.5

Module Components

Number	Name	Type	CP
CH-222-A	Mathematical and Physical Foundations of Robotics II	Lecture	5
CH-222-B	Mathematical and Physical Foundations of Robotics II - Lab	Lab	2.5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr. Francesco Maurelli	• Robotics and Intelligent Systems (RIS)	Mandatory for RIS and minor RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Lecture (35 hours) • Lab (17.5 hours) • Private study (115 hours) • Exam preparation (20 hours)
☒ None	☒ None			
			Duration	**Workload**
			1 semester	187.5 hours

Recommendations for Preparation
Review basic mathematical concepts.

Content and Educational Aims

The course offers an introduction to modeling, and design of linear systems as they appear in the context of robotics, in particular including important introductory aspects of control systems. We discuss these topics in terms of ordinary differential equations (odes). Students learn how to analyze and solve systems of such odes using state and frequency space methods. Concepts covered include time and frequency response, stability, steady-state errors, and the root-locus method. The ability to design feedback control systems, in particular the famous PID controller, is the main goal of the course. Practical experience will be gained based on numerous examples, analysis and design tasks supported by numerical methods. The lab is designed to guide students through practical hands-on work with various components of intelligent systems. It will focus on the interfacing of a microcontroller with commonly used sensors and actuators.

Intended Learning Outcomes

By the end of this module, successful students will be able to

1. Model simple common mechanical and electrical systems which are part of robots
2. Solve linear systems and tune their behavior in simulation and analytically
3. Understand and be able to apply the uni-lateral Laplace transforms and its inverse
4. Understand basic approaches and methods related to feedback control systems
5. Create an electromechanical model of a brushed DC motor in Simulink and study its properties;
6. Design and tune PID controllers for motor-speed control and for servo control;
7. Present and justify their work appropriately in accordance with scientific standards.

Indicative Literature

Norman S. Nise (2012), Control Systems Engineering

Usability and Relationship to other Modules

- This module is the foundation of the CORE modules in the following years.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Module achievement: Lab report

Completion: To pass this module, the examination has to be passed with at least 45%.

7.3 Programming in C and C++

Module Name Programming in C and C++	**Module Code** CH-230	**Level (type)** Year 1 (CHOICE)	**CP** 7.5

Module Components

Number	Name	Type	CP
CH-230-A	Programming in C and C++	Lecture	5
CH-230-B	Programming in C and C++ - Tutorial	Tutorial	2,5

Module Coordinator Dr. Kinga Lipskoch	**Program Affiliation** • Computer Science (CS)	**Mandatory Status** Mandatory for CS, SDT, RIS, ECE, minor CS, minor RIS and minor Software Development

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills
☒ None	☒ None	

Frequency Annually (Fall)	**Forms of Learning and Teaching** • Lecture attendance (35 hours) • Tutorial attendance (17.5 hours) • Independent study (115 hours) • Exam preparation (20 hours)
Duration 1 semester	**Workload** 187.5 hours

Recommendations for Preparation

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.

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students will be able to

1. explain basic concepts of imperative programming languages such as variables, assignments, loops, and function calls;
2. write, test, and debug programs in the procedural programming language C using basic C library functions;
3. demonstrate how to use pointers to create dynamically allocated data structures such as linked lists;
4. explain the relationship between pointers and arrays;
5. illustrate basic object-oriented programming concepts such as objects, classes, information hiding, and inheritance;
6. give original examples of function and operator overloading and polymorphism;
7. 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.

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.

Examination Type: Module Component Examinations Component

1: Lecture

Assessment types: Written examination Duration: 120 min

Weight: 67%
Scope: All theoretical intended learning outcomes of the module

Component 2: Tutorial

Assessment: Program Code

Weight: 33%

Scope: All practical intended learning outcomes of the module

Completion: To pass this module, the examination of each module component has to be passed with at least 45%.

7.4 General Electrical Engineering I

Module Name General Electrical Engineering I	**Module Code** CH-210	**Level (type)** Year 1 (CHOICE)	**CP** 7.5

Module Components			
Number	Name	Type	CP
CH-210-A	General Electrical Engineering I	Lecture	5.0
CH-210-B	General Electrical Engineering Lab I	Lab	2.5

Module Coordinator Prof. Dr. Giuseppe Abreu	**Program Affiliation** • Electrical and Computer Engineering (ECE)	**Mandatory Status** Mandatory for ECE, minor ECE and RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall)	• Lecture (35 hours) • Lab (25.5 hours) • Private Study (127)
☒ None	☒ None	• Basic mathematics, including notions of vectors, matrices functions, and complex numbers	**Duration** 1 semester	**Workload** 187.5 hours

Recommendations for Preparation

It is highly recommended that students familiarize themselves with the contents of the appendices of a typical introductory textbook on Electrical Engineering (e.g. “Fundamentals of Electric Circuits”, by Alexander and Sadiku and “Basic Engineering Circuit Analysis”, by Irwin and Nelms), including Complex Numbers and basic Linear Algebra (in particular the solution of simultaneous linear equations). In addition, it is recommended that students acquire Calculus basics (differentiation and integration of simple functions).

Content and Educational Aims

The module, consisting of a lecture, supported by corresponding lab experiments, comprises the classical introduction to Electrical and Computer Engineering (ECE), starting from the basics of the electric phenomenon, its fundamental elements (charge, current, potential, energy, etc.), its interaction with materials (conductivity, capacitance, inductance, etc.) and its manipulation by man-made structures (electronic components and circuits). The module then develops into a wide set of general principles, laws and analytical tools to understand electric circuits and electric systems in general. The module also offers a solid foundation on which specialization areas in EE (e.g. Communications, Control, etc.) are built. The emphasis is the analysis of circuits in DC steady state and transient modes. Classic material include (but are not limited to): Kirchhoff’s Laws, Volta’s Law (capacitance), Faraday’s Law (inductance), Thevenin and Norton’s Theorem, Tellegen’s Theorem, delty-wye transformation, source transformations, basics of non-linear electronic components (diodes and transistors), OpAmp circuits, State-space Method, Laplace Transform applied to the analysis of higher-order circuits, Laplace impedances and transfer functions. In the lab portion of the module, users will familiarize themselves with electronic components (resistors, capacitors, inductors, diodes, OpAmps, transistors, etc.) and circuits, and learn how to utilize typical lab equipment (such as breadboards, digital multimeters, voltage and current sources and function generators) required for the assembly and analysis of electric circuits.

Intended Learning Outcomes

By the end of this module, students should be able to

1. describe the fundamental physical principles of electric quantities (charge, current, potential, energy and its conservation, etc.);
2. explain how the aforementioned quantities relate to each other and interact with matter, including corresponding mathematical models;
3. explain how the aforementioned models can be utilized to manipulate electric quantities and phenomenon in the form of electric and electronic circuits or machines that perform several tasks and functions according to intended designs;
4. employ various theoretical and practical tools to analyze electric circuits including resistive circuits, reactive circuits, and OpAmp circuits, both in DC steady-state and transient modes.

In addition to the aforementioned outcomes, fundamental to a career in ECE, students will also have acquired:

5. analytical and mathematical modeling skills useful to study other physical systems (e.g. in other areas of Engineering, Physics, Robotics, etc.)
6. the ability to work in a lab environment and operate lab equipment, as required in other professions (e.g. Physics, Biology, Chemistry etc.).

Usability and Relationship to other Modules

Indicative Literature

Charles K. Alexander and Matthew N. O. Sadiku, Fundamentals of Electric Circuits, 3rd ed., McGraw-Hill, 2008 (Primary Textbook).

J. David Irwin and R. Mark Nelms, Basic Engineering Circuit Analysis, 10th ed., Wiley, 2010 (Recommended Reference).

James Nilsson and Susan Riedel, Electric Circuits, 10th ed., Pearson, 2015 (Extra Reference).

A. Agarwal and J. Lang, Foundations of Analog and Digital Electronic Circuits, 1st ed., Elsevier, 2005 (Advanced Reference for selected topics).

Examination Type: Module Component Examinations

Module Component 1: Lecture

Assessment Type: Written examination	Duration: 120 min Weight: 67%
Scope: Intended learning outcomes of the lecture (1-3,5)	

Module Component 2: Lab

Assessment Type: Laboratory reports	Length: 5-10 pages per experiment session Weight: 33%
Scope: Intended learning outcomes of the lab (3-4, 6).	

Completion: To pass this module, the examination of each module component has to be passed with at least 45%.

7.5 Algorithms and Data Structures

Module Name Algorithms and Data Structures	**Module Code** CH-231	**Level (type)** Year 1 (CHOICE)	**CP** 7.5

Module Components			
Number	Name	Type	CP
CH-231-A	Algorithms and Data Structures	Lecture	7.5

Module Coordinator Dr. Kinga Lipskoch	**Program Affiliation** • Computer Science (CS)	**Mandatory Status** Mandatory for CS, RIS, and minor in CS

Entry Requirements			**Frequency** Annually (Spring)	**Forms of Learning and Teaching** • Class attendance (52.5 hours) • Independent study (115 hours) • Exam preparation (20 hours)
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills		
☒ Programming in C and C++	☒ None		**Duration** 1 semester	**Workload** 187.5 hours

Recommendations for Preparation

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.

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students will be able to

1. explain asymptotic (time and memory) complexities and respective notations;
2. able to prove asymptotic complexities of algorithms;
3. illustrate basic data structures such as arrays, lists, queues, stacks, trees, and hash tables;
4. describe algorithmic design concepts and apply them to new problems;
5. explain basic algorithms (sorting, searching, graph algorithms, computational geometry) and their complexities;
6. 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.

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.

Examination Type: Module Examination

Assessment Type: Written examination Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%

7.6 Digital Systems and Computer Architecture

Module Name Digital Systems and Computer Architecture	**Module Code** CH-234	**Level (type)** Year 1 (CHOICE)	**CP** 7.5

Module Components

Number	Name	Type	CP
CH-234-A	Digital Systems and Computer Architecture	Lecture	5.0
CH-234-B	Digital Systems and Computer Architecture Tutorial	Tutorial	2.5

Module Coordinator Prof. Dr. Jürgen Schöwälder	**Program Affiliation** • Computer Science (CS)	**Mandatory Status** Mandatory for CS, RIS and ECE Mandatory elective for SDT

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Lecture attendance (35 hours) • Tutorial attendance (17.5 hours) • Independent study (115 hours) • Exam preparation (20 • hours)
☒ None	☒ None		**Duration** 1 semester	**Workload** 187.5 hours

Recommendations for Preparation

Content and Educational Aims

The module introduces the essential hardware components of a digital computer system. Students will learn how useful digital circuits to add numbers or to store data can be constructed out of basic logic gates. Using these building blocks, the module will introduce how a simple processor can be constructed and how it interacts with memory systems and other components of a computer system. Students will practice the basics of assembler programming to understand program execution at the hardware level.

Intended Learning Outcomes

By the end of this module, students will be able to

1. Understand the architecture of a digital computer;
2. explain the representation of numbers (integers and floats);
3. summarize basic laws of Boolean algebra;
4. describe basic logic gates and which Boolean functions they implement;
5. construct and analyze basic combinational digital circuits (e.g., adder, comparator, multiplexer);
6. design and analyze basic sequential digital circuits (e.g., latches, flip-flops);
7. outline the basic structure of the von Neumann computer architecture;
8. explain the execution of machine instructions on a von Neumann computer;
9. develop simple programs in an assembler language such as the RISC-V;
10. demonstrate how function calls are executed and the role of the stack;
11. understand microarchitectural concepts and the importance of the memory hierarchy;
12. explain the purpose and principles of operation of the components of a computer system.

Indicative Literature

- John L Hennessy, David A. Patterson: Computer Architecture: A Quantitative Approach, 6th edition, Morgan Kaufmann, 2017
- Sarah Harris, David Harris: Digital Design and Computer Architecture: RISC-V Edition, Morgan Kaufmann, 2021

Usability and Relationship to other Modules

This module introduces students to the digital hardware components of a computer system. Students attain an understanding of program execution at the hardware level. Other modules requiring an understanding of program execution at the hardware level may require this module as a prerequisite.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Module achievement: 50% of ten weekly assignments correctly solved. Two additional assignments are offered during the semester and another assignment is offered in August to makeup missing points.

Completion: To pass this module, the examination has to be passed with at least 45%.

7.7 Robotics

Module Name Robotics		**Module Code** CO-540	**Level (type)** Year 2 (CORE)	**CP** 5
Module Components				
Number	Name		Type	CP
CO-540-A	Robotics		Lecture	5
Module Coordinator Prof. Dr. Andreas Birk	**Program Affiliation** • Robotics and Intelligent Systems (RIS)		**Mandatory Status** Mandatory for RIS and minor RIS Mandatory elective for CS	
Entry Requirements Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills **Frequency** Annually (Fall)	**Forms of Learning and Teaching** • Class attendance (35	

☒ Programming in C/C++ ☒ Mathematical and Physical Foundations of Robotics I	☒ None		hours) - Private study (70 hours) - Exam preparation (20 hours)
		Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

Revise content of the pre-requisite modules.

Content and Educational Aims

Robotics is an area that is driven by dreams from science fiction and the reality of engineering. The module intends to provide an understanding of the formal foundations of this area as well as its technological state of the art and future directions. The course accordingly gives an introduction to the core algorithmic, mathematical, and engineering concepts and methods of robotics. This includes concepts and methods that are used for well- established tools of factory automation, especially in the form of robot-arms, as well as increasingly relevant intelligent mobile systems such as autonomous cars or autonomous transport systems.

Intended Learning Outcomes

By the end of this module, students should be able to

1. outline and explain the history, general developments, and application areas of robotics;
2. apply the concepts and methods to describe space and motions therein including homogeneous coordinates and transforms as well as quaternions;
3. use the spatial concepts and methods for the forward kinematics (FK) of robot-arms;
4. explain basic concepts of simple actuators, including electrical motors and gear systems;
5. apply concepts and methods to derive the inverse kinematics of robot-arms and related systems such as legs in analytical and numerical forms;
6. apply concepts and methods of wheeled locomotion including FK and IK of the differential and of the omni-directional drive;
7. use basic concepts and methods of dynamics;
8. Explain and use core concepts and methods of global localization, e.g., multilateration and multidimensional scaling;
9. use the basic concepts and methods of error propagation estimation in the context of relative localization with dead-reckoning;
10. outline and compare the basic concepts and methods of mapping.

Indicative Literature

J. J. Craig, Introduction to robotics - Mechanics and control, Prentice Hall, 2005.

G. Dudek and M. Jenkin, Computational Principles of Mobile Robotics, Cambridge University Press, 2000.

R. Siegwart and I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, The MIT Press, 2004.

S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, MIT Press, 2005.

H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, Principles of Robot Motion, MIT Press, 2005.

Usability and Relationship to other Modules

- This module serves as a third Year Specialization module for CS major students.
- This module gives an introduction to Robotics, which is a core discipline of Robotics and Intelligent System (RIS) and an important area of possible future employment.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min

Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.8 Machine Learning

Module Name	**Module Code**	**Level (type)**	**CP**
Machine Learning	CO-541	Year 2 (CORE)	5

Module Components

Number	Name	Type	CP
CO-541-A	Machine Learning	Lecture	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr. Francesco Maurelli	• Robotics and Intelligent Systems (RIS)	Mandatory for RIS, MMDA, , SDT, CS and minor in Software Development Mandatory elective for PHDS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)
☒ None	☒None	• Knowledge and command of probability theory and methods, as in the module “Probability and Random Process”		
			Duration	**Workload**
			1 semester	125 hours

Recommendations for Preparation

None

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students should be able to

1. understand the notion of probability spaces and random variables;
2. understand basic linear modeling and estimation techniques;
3. understand the fundamental nature of the “curse of dimensionality;”
4. understand the fundamental nature of the bias-variance problem and standard coping strategies;
5. use elementary classification learning methods (linear discrimination, radial basis function networks, multilayer perceptrons);

6. implement an end-to-end learning suite, including feature extraction and objective function optimization with regularization based on cross-validation.

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.

Usability and Relationship to other Modules

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

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min

Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.9 RIS Lab

Module Name RIS Lab	**Module Code** CO-542	**Level (type)** Year 2 (CORE)	**CP** 5

Module Components

Number	Name	Type	CP
CO-542-A	RIS Lab I	Lecture/lab	2.5
CO-542-B	RIS Lab II	Lecture/lab	2.5

Module Coordinator Prof. Dr. Francesco Maurelli	**Program Affiliation** • Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory minor RIS Mandatory elective for RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall)	• Class attendance (35 hours) • Private study (70 hours) • Report preparation (20 hours)
☒ Programming in C/C++	☒ None		**Duration** 2 semesters	**Workload** 125 hours

Recommendations for Preparation

None

Content and Educational Aims

RIS Lab I focuses on robotics middleware such as the Robot Operating System (ROS). Building on the programming class and on the introductory course, it presents ways in which different units of a robotic system can share information. The work will be mainly in simulation, using the ROS Gazebo package or similar.

RIS Lab II brings the students to apply what they learned in simulation with real robotics systems. Interacting with the physical world adds a layer of complexity which students will face for the first time. They will learn the differences between working in simulation and working with real systems, will learn how to integrate software to run real time and to articulate and describe their journey in a scientific manner.

Students are also introduced to and practice technical and scientific writing skills in preparation for their thesis.

Intended Learning Outcomes

By the end of this module, students should be able to

1. describe robotics software architecture;
2. correctly use available libraries and packages;
3. create new packages and functionalities in a robotics simulator;
4. create new packages and functionalities using real robotics systems
5. develop team spirit collaborating with peers to access the robotics systems

Indicative Literature A. Koubaa, Robot Operating System (ROS), The Complete Reference Vol 1, Springer, 2018.
Usability and Relationship to other Modules
Examination Type: Module Component Examination Module **Component 1: Lab 1** Assessment Type: Laboratory report Length: approx. 10 pages Weight: 50% Scope: Intended learning outcomes of RIS Lab I - 1, 2, 3, 6. **Module Component 2: Lab 2** Assessment Type: Laboratory report Length: approx. 10 pages Weight: 50% Scope: Intended learning outcomes of RIS Lab II - 4, 5. Completion: To pass this module, both module component examinations have to be passed with at least 45%.

7.10 Automation

Module Name	Module Code	Level (type)	CP
Automation	CO-543	Year 2 (CORE)	5

Module Components

Number	Name	Type	CP
CO-543-A	Automation	Lecture	5

Module Coordinator	Program Affiliation	Mandatory Status
Prof. Dr. Francesco Maurelli	• Robotics and Intelligent Systems (RIS)	Mandatory elective for RIS

Entry Requirements			Frequency	Forms of Learning and Teaching
Pre-requisites ☒ Programming C/C++ ☒ Mathematical and Physical Foundations of Robotics I	Co-requisites ☒ None	Knowledge, Abilities, or Skills • Understanding of the basics of electronics • Calculus • basic C/C++/Python • basic MATLAB/Simulink or SciLab	Annually (Spring)	• Lectures (30 hours) • Lab (5 hours) • Private study (70 hours) • Exam preparation (20 hours)
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

Review material of Embedded Systems Lab.

Content and Educational Aims

Automation is the application of science and technology to control mechanical systems, including situations in which this proposed solution duplicates the skills of a human operator or even exceeds them. Industrial automation concentrates on solutions in the production and delivery of products and services.

The field of automation has considerable overlap with the fields of Control and Robotics. However, the distinguishing aspect is the emphasis on an industrial performance and setting, along with the concomitant focus on robustness and efficiency under factory conditions.

The topics covered in this course include: an introduction to sensors and their scientific principles; filtering, data fusion and estimation; types of actuators and details about the operation of industrial motors and drives; an introduction to programmable logic controllers (PLCs); their hierarchy and different PLC programming paradigms; and artificial intelligence (AI) concepts used in automation, such as state machines and sensor data processing.

Intended Learning Outcomes

By the end of this module, students should be able to

1. explain the characteristics and principles of a number of industrial sensors and electric motors, comment on their overall parameters such as accuracy and precision, and outline the reasons for the calibration process;
2. apply this knowledge to translate simple machine specifications into an automation problem in terms of sensing, actuation, and processing strategy at the conceptual level, including an educated selection of sensors and drives;

3. apply a family of filtering and estimation techniques covered in the lectures to systems similar to those used in the examples; recall the analysis of their stability and duplicate it in the case of the presented system;
4. apply the state machine concept to simple processes and routines;
5. explain the strengths, principles, and programming paradigms of PLCs;
6. recall the currently used concept in organizing a factory-wide automation pyramid and understand the working of at least one automation communication protocol in detail;
7. combine the skills mentioned above in proposing solutions to simple industrial problem examples.

.

Indicative Literature

N. Zuech, Handbook of Intelligent Sensors for Industrial Automation, Addison-Wesley, 1992.
A. Hughes, Electric Motors and Drives, 3rd edition, 2006.
K. Collins, PLC Programming for Industrial Automation, 2007.

Usability and Relationship to other Modules

- A portion of the knowledge is complementary with the Control Systems course
- The robotics course completes the information given in this course with respect to mobile machinery.

Examination Type: Module Examination

Assessment Type: Written examination Duration: 150 min
Weight: 100%

Scope: The course material excluding programming skills.

The exam will provide a number of multiple choice of true/false questions, where students will be expected to recall facts and principles covered in the class.
Sample problems will be given, similar to those given in class, where the students will be expected to duplicate the calculations and choice principles explained in the class.
An open-ended question will test their understanding of the entire concepts such as calibration or state machine.

Completion: To pass this module, the examination has to be passed with at least 45%.

7.11 Embedded Systems

Module Name Embedded Systems	**Module Code** CO-544	**Level (type)** Year 2 (CORE)	**CP** 5

Module Components

Number	Name	Type	CP
CO-544-A	Embedded Systems	Lecture/Lab	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Dr. Fangning Hu	• Robotics and Intelligent Systems (RIS)	Mandatory elective for RIS

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	**Frequency**	**Forms of Learning and Teaching**
☒ Programming in C/C++	☒ None		Annually (Fall)	• Lecture/Lab (35 hours) • Private study (90 hours)
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

Revising programming in C and the binary number systems.

Content and Educational Aims

Microcontrollers are core components of modern devices. Designed to handle sensor data and to control actuators, equipped with considerable computational power at relatively low cost and with limited power consumption, they are enablers of our rapidly growing technological environment, in particular, when it comes to mobile systems. We are going to use the AVR/ARM processor based on the RISC-architecture, which is becoming increasingly popular with its use in smartphones, tablets, and various forms of embedded systems, owing to its small size and low power consumption. The course provides a sound introduction to these nearly ubiquitous devices and guides the students in an application-oriented manner through a series of design tasks. The list of topics includes the basic architecture of a microcontroller with its ALU, timer/counter, memory, and I/O interface; the concepts of working registers, interrupt vectors, and program counters; necessary programming tools such as embedded C and assembler, as well as several implementation problems such as reading/controlling various sensors/actuators, processing internal/external interrupts, generation of PWM signals, and AD/DA conversion. At the end of the course, students should be able to develop and implement their own solutions for typical applications on
AVR/ARM-based microcontrollers.

Intended Learning Outcomes

By the end of this module, students should be able to

1. describe the architecture of a microcontroller;
2. understand the datasheet of a microcontroller;
3. program a microcontroller to read/control sensors/actuators, process interrupters, generate PWM, and perform AD/DA conversion;
4. design a solution for an embedded application by microcontroller.

Indicative Literature

Online resources and manuals provided by the Instructor of Records.

M. Michalkiewics et. al, AVR C Runtime Library, http://savannah.nongnu.org/projects/avr-libc/, accessed 3 March 2020.

Usability and Relationship to other Modules

- This module introduces the architecture of an AVR/ARM-based microcontroller and how to program it. It could also serve as a specialization course for students from Electrical and Computer Engineering and Computer Science.

Examination Type: Module Examination

Assessment Type: Project Assessment

Duration: 180 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.12 Control Systems

Module Name Control Systems	**Module Code** CO-545	**Level (type)** Year 2 (CORE)	**CP** 5

Module Components

Number	Name	Type	CP
CO-545-A	Control Systems	Lecture	5

Module Coordinator Prof. Dr. Francesco Maurelli	**Program Affiliation** • Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory Elective for RIS

Entry Requirements			**Frequency** Annually (Fall)	**Forms of Learning and Teaching** • Lecture (35 hours) • Private study (90 hours)
Pre-requisites ☒ Elements of Linear Algebra ☒ Elements of Calculus (or the Matrix version) ☒ Mathematical and Physical Foundations of Robotics II	Co-requisites ☒ None	Knowledge, Abilities, or Skills • Transfer functions • Laplace transforms	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

Revise calculus, linear algebra, Laplace transforms, and obtain the course textbook in advance of the first class. Please see course pages for details.

Content and Educational Aims

This course offers a systematic walk through the fundamentals of control theory for linear systems. Building on the introduction to RIS course, new concepts, perspectives, and skills will be introduced and discussed. In particular, this includes (different) state space representations, reduction techniques for larger block diagrams, the BIBO perspective on stability, the role of disturbances, and the related question of sensitivity. We will also study new approaches to improve the response of a given system via lead and lag compensators, including feedback techniques. The major new analytic tools will be the Nyquist plot and techniques based on it.

Intended Learning Outcomes

By the end of this course, successful students will be able to

1. understand and apply fundamental concepts from linear control theory;
2. reduce larger block diagrams;
3. use various methods (Routh table, root locus, Nyquist) to analyze systems for stability;
4. find the steady-state errors for various standard input signals;
5. understand and quantify the sensitivity of steady-state errors with regard to parameter deviations;
6. design lead and lag compensators to improve the system response.

Indicative Literature

N.S. Nise: Control Systems Engineering, John Wiley & Sons, 2010.

Usability and Relationship to other Modules

This module introduces the students to the field of automatic control and is strongly related to the embedded systems, automation, and robotics modules. However, it also helps to better understand how systems in general, be they mechanical, electrical, biological, or even social, such as smart cities, can be maintained under stable conditions and with desired response characteristics.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.13 Computer Vision

Module Name Computer Vision	**Module Code** CO-546	**Level (type)** Year 2 (CORE)	**CP** 5
Module Components			
Number	Name	Type	CP
CO-546-A	Computer Vision	Lecture/lab	5

Module Coordinator Prof. Dr. Francesco Maurelli	**Program Affiliation** • Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory elective for RIS, CS

Entry Requirements			**Frequency** Annually (Fall)	**Forms of Learning and Teaching** • Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)
Pre-requisites ☒ Mathematical and Physical Foundations of Robotics I ☒ Programming in C/C++	Co- requisites ☒ None	Knowledge, Abilities, or Skills • Basic knowledge of robotics middleware (RIS Lab I)		
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

Refresh basic programming skills in MATLAB and/or Python

Content and Educational Aims

Computer Vision algorithms are used in a variety of real-world applications that include surveillance and object tracking, 3D model building (photogrammetry), and object recognition. Apart from their visual appeal, these algorithms also represent elegant applications of linear algebra and optimization techniques. Topics covered in this course include a recapitulation of relevant linear algebra, introduction to face-recognition, camera calibration, stitched panoramas, edge and blob visual features, structure from motion, color-spaces, segmentation, and an introduction to object-recognition.

Intended Learning Outcomes

By the end of this module, students should be able

1. describe image formation and camera models;
2. calibrate cameras;
3. compute image histograms, and basic image processing;
4. discriminate among visual features (e.g., corner, edge, blob);
5. Properly use computer vision libraries;
6. implement computer vision applications.

Indicative Literature

D.A. Forsyth and J. Ponce, Computer Vision: A Modern Approach. 2nd edition, 2011.

R. Szeliski, Computer Vision: Algorithms and Applications, Springer, http://szeliski.org/Book, 2010. Ma et

al., An Invitation to 3 D Vision: From Images to Geometric Models, Springer, 2004.

Usability and Relationship to other Modules

- Giving the foundation of computer vision, this module is important for RIS project and for advanced specialization courses.
- This module serves as a third year Specialization module for CS major students.
- This module belongs to the Data Science Track in the MSc AST

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min

Weight: 100%

Scope: All intended learning outcomes of the module

Module achievements: 50% if the assignments correctly solved

Completion: To pass this module, the examination has to be passed with at least 45%.

7.14 Artificial Intelligence

Module Name Artificial Intelligence	**Module Code** CO-547	**Level (type)** Year 2 (CORE)	**CP** 5

Module Components

Number	Name	Type	CP
CO-547-A	Artificial Intelligence	Lecture	5

Module Coordinator Prof. Dr. Andreas Birk	**Program Affiliation** • Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory for RIS and minor in RIS Mandatory elective for CS and SDT

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)
☒ Algorithms and data structures or Core Algorithms and Data Structures	☒ None		**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

Revise content of the pre-requisite modules.

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students should be able to

1. outline and explain the history, general developments, and application areas of AI;
2. apply the basic concepts and methods of behavior-oriented AI;
3. use concepts and methods of search algorithms for problem-solving;
4. explain the basic concepts of path-planning as an application example for domain-specific search;
5. apply basic path-planning algorithms and to compare their relations to general search algorithms;
6. write and explain concepts of propositional and first-order logic;
7. use logic representations and inference for basic examples of artificial planning systems.

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.

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.

Examination Type: Module Examination

Assessment Type: Written examination	Duration: 120 min Weight: 100%
Scope: All intended learning outcomes of the module	

Completion: To pass this module, the examination has to be passed with at least 45%.

7.15 RIS Project

<table>
<tr><td colspan="3"><b>Module Name</b>
RIS Project</td><td><b>Module Code</b>
CO-548</td><td><b>Level (type)</b>
Year 2 (CORE)</td><td><b>CP</b>
5</td></tr>
<tr><td colspan="6"><b>Module Components</b></td></tr>
<tr><td>Number</td><td colspan="3">Name</td><td>Type</td><td>CP</td></tr>
<tr><td>CO-548-A</td><td colspan="3">RIS Project</td><td>Lecture/lab</td><td>5</td></tr>
<tr><td><b>Module Coordinator</b>
Prof. Dr. Francesco Maurelli</td><td colspan="3"><b>Program Affiliation</b>
• Robotics and Intelligent Systems (RIS)</td><td colspan="2"><b>Mandatory Status</b>
Mandatory for RIS</td></tr>
<tr><td colspan="3"><b>Entry Requirements</b></td><td rowspan="3"><b>Frequency</b>
Annually (Spring)</td><td colspan="2" rowspan="3"><b>Forms of Learning and Teaching</b>
• Class attendance (35 hours)
• Private study (70 hours)
• Report preparation (20 hours)</td></tr>
<tr><td>Pre-requisites</td><td>Co-requisites</td><td>Knowledge, Abilities, or Skills</td></tr>
<tr><td rowspan="2">☒ Mathematical and Physical Foundations of Robotics I
☒ Programming in C/C++</td><td rowspan="2">☒ None</td><td rowspan="2">• Basic knowledge of robotics middleware (RIS Lab I)</td></tr>
<tr><td><b>Duration</b>
1 semester</td><td colspan="2"><b>Workload</b>
125 hours</td></tr>
<tr><td colspan="6"><b>Recommendations for Preparation</b>
None</td></tr>
</table>

Content and Educational Aims

The aim of RIS project is to use real robotics systems (e.g., Duckietown for autonomous driving) to design and implement a project that is related to one or more modules of the RIS program. Students will work in groups and will choose a scenario to focus on, involving a combination of robotics, computer vision, machine learning, artificial intelligence, and control systems competences. The lecture part of the module will focus on the transition from work in simulation to work with real robotics systems, including basic health and safety procedures.

Intended Learning Outcomes

By the end of this module, students should be able to

1. apply available libraries to real robotics systems;
2. develop new robotics functionalities;
3. integrate new functionalities in robotics systems;
4. design and plan a project over several weeks;
5. work in a team, overcoming challenges;
6. present scientific results in an adequate manner.

Indicative Literature

Not specified

Usability and Relationship to other Modules

- This module represents a glue among various different core modules, focusing on the design and implementation of a project with real robotics systems. It is pivotal for advanced courses in the third year and lays the foundation for the competence skills required for the thesis.

Examination Type: Module Examination

Assessment Component 1: Project report	Length: approx. 15 pages Weight: 75%
Scope: Intended learning outcomes of the lecture 1, 2, 3, 4, 5).	
Assessment Component 2: Presentation	Duration: approx. 15 min Weight: 25%
Scope: Intended learning outcomes of the lab 4, 5, 6.	

Completion: This module is passed with an assessment-component weighted average grade of 45% or higher.

7.16 Marine Robotics

Module Name	Module Code	Level (type)	CP
Marine Robotics	CA-S-RIS-801	Year 3 (Specialization)	5

Module Components

Number	Name	Type	CP
CA-RIS-801	Marine Robotics	Lecture/lab	5

Module Coordinator	Program Affiliation	Mandatory Status
Prof. Dr. Francesco Maurelli	• Robotics and Intelligent Systems (RIS)	Mandatory Elective for RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)
☒ Mathematical and Physical Foundations of Robotics I ☒ Programming in C/C++	☒ None	• Basic knowledge of robotics middleware (RIS Lab I)	**Duration**	**Workload**
			1 semester	125 hours

Recommendations for Preparation

None

Content and Educational Aims

Marine robotics currently plays a key role in the exploitation of marine resources (offshore), conservation of marine environments (environment assessment), and security applications (harbor protection). The European Commission has estimated that the economic impact of the “blue” economy, which considers all activities linked to the sea, is worth more than €400 billion annually, with more than €150 billion in activities directly related to marine activities.

This module builds on the CORE courses of the second year with a specialization on (intelligent) marine robotics, studying the typical environmental constraints, technical solutions, and current trends.

The topics covered by this module include ROV and AUV operations, underwater acoustic, underwater sensing, navigation, communication, and multivehicle cooperation.

The module will have a practical component, with the possibility of visiting nearby institutions and participating in field excursions.

Intended Learning Outcomes

By the end of this module, students should be able to

1. understand the challenges in the marine domain for robotics systems;
2. analyze the functioning of acoustic devices for robot autonomy;
3. develop advanced functionalities for a marine robot in a simulation;
4. develop advanced functionalities for a marine robot in the field.

Indicative Literature

L. Jaulin et. al, Marine Robotics and Applications , Springer, 2018.

S. W. Moore, Underwater Robotics: Science, Design & Fabrication, 2010.

B. Siciliano O. Khatib, Springer Handbook of Robotics, Springer, 2008.

Usability and Relationship to other Modules

- This module is a robotics-oriented specialization course, with the possibility to work with real robots.

Examination Type: Module Examination

Assessment Type: Oral examination

Duration: approx. 15 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.17 Human-Robot Interaction

Module Name Human-Robot Interaction			**Module Code** CA-S-RIS-804	**Level (type)** Year 3 (Specialization)	**CP** 5
Module Components					
Number	Name			Type	CP
CA-S-RIS-804	Human-Robot Interaction			Lectures	5
Module Coordinator Prof. Dr. Jakob Suchan	**Program Affiliation** Robotics and Intelligent Systems (RIS)			**Mandatory Status** Mandatory Elective for RIS	
Entry Requirements Module pre-requisites	Module co-requisites	Recommended knowledge, abilities, or skills	**Frequency** Annually (Fall)	**Forms of Learning and Teaching** • Class attendance (35 hours) • Private study (70hours) • Exam preparation (20 hours)	

☒ None	☒ None	None	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

None

Content and Educational Aims

Human-robot interaction (HRI) is concerned with the problem of how to design efficient and natural interaction between humans and robots in both everyday tasks and professional contexts. This course introduces students to the key disciplines behind HRI and provides a multidisciplinary view includes elements of computer science, robotics, psychology, sociology, and design as they relate to HRI. The course explores the components and control of robotic systems, key interaction modalities (spatial, verbal, non-verbal, and emotional), and their role in creating effective human-robot interactions. Students will explore current challenges in HRI, evaluate solution strategies, and examine real-world and emerging applications of robots and autonomous systems in society.

Intended Learning Outcomes

After completing this module, students will be able to:

1. Explain interdisciplinary aspects of HRI.
2. Evaluate different types of robots / autonomous systems.
3. Explain the implications of the robot design for the way they interact with humans.
4. Describe key Challenges for robots in spatial interaction.
5. Identify and discuss different interaction/ communication modalities (verbal, nonverbal).
6. Explain psychological Aspects of Human Robot Interaction.
7. Design and conduct basic human-behavioral experiments and analyze the data.
8. Implement interaction strategies for autonomous assistive systems.
9. Explain ethical implications of autonomous systems and future developments in HRI.

Indicative Literature

Bartneck C, Belpaeme T, Eyssel F, Kanda T, Keijsers M, Šabanović S. *Human-Robot Interaction: An Introduction*. Cambridge University Press; 2020.

Usability and Relationship to other Modules

This module provides a comprehensive introduction to Human-Robot Interaction, it complements the Robotics module, which provides a deeper understanding of the basics of robotic systems, and the Artificial Intelligence module, which gives an introduction to classical AI methods and agent-based approaches to AI.

Examination Type: Module Component Examination

Assessment Type: Project Assessment Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.18 Human Computer Interaction

Module Name Human Computer Interaction	**Module Code** ACS-306	**Level (type)** Year 3 (Specialization)	**CP** 5

Module Components

Number	Name	Type	CP
ACS-306	Human Computer Interaction	Lecture	5

Module Coordinator Prof. Dr. Francesco Maurelli	**Program Affiliation** • Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory elective for RIS, ACS

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills
☒ None	☒ None	• None

Frequency Annually (Fall)	**Forms of Learning and Teaching** • Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)	
Duration 1 semester	**Workload** 125 hours	

Recommendations for Preparation

None

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students should be able to

1. explain the evolution of human–computer interaction models;
2. design and implement simple graphical user interfaces;
3. explain ergonomic principles guiding the design of user interfaces;
4. illustrate different types of interaction (e.g., visual, voice, gestures) and their usability aspects;
5. evaluate aspects of and tradeoffs between usability, learnability, efficiency, and safety;
6. apply scientific methods to evaluate interfaces with respect to their usability and other desirable properties;
7. use prototyping tools that can be employed to create mockups of user interfaces during the early stages of a software project.

Indicative Literature

Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale: Human-Computer Interaction, 3rd edition, Pearson, 2004

Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th edition, Pearson, 2016

Céline Jost, Brigitte Le Pévédic, Tony Belpaeme, Cindy Bethel, Dimitrios Chrysostomou, Nigel Crook, Marine Grandgeorge, Nicole Mirnig, Human-Robot Interaction, Evaluation Methods and Their Standardization, Springer 2020 ISBN: 978-3-030-42306-3

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.

Examination Type: Module Examination

Assessment Type: Project assessment	Duration: 120 min Weight: 100%
Scope: All intended learning outcomes of the module	

Completion: To pass this module, the examination has to be passed with at least 45%.

7.19 Optimization

Module Name Optimization	**Module Code** CA-S-RIS-803	**Level (type)** Year 3 (Specialization)	**CP** 5

Module Components

Number	Name	Type	CP
CA-RIS-803	Optimization	Lecture	5

Module Coordinator Prof. Dr. Francesco Maurelli	**Program Affiliation** • Robotics and Intelligent Systems (RIS)	**Mandatory Status** Mandatory elective for RIS

Entry Requirements Pre-requisites ☒ Elements of Linear Algebra ☒ Elements of Calculus (or the Matrix version)	Co- requisites ☒ None	Knowledge, Abilities, or Skills	**Frequency** Annually (Spring)	**Forms of Learning and Teaching** • Lecture (35 hours) • Private study (90 hours)
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

Revise calculus and linear algebra from your first year.

Content and Educational Aims

Optimization is a key step in the design of systems and processes. The course starts with a review of multidimensional calculus applied to unconstrained problems. It then focuses on equality- and inequality- constrained cases from the perspective of the Lagrange formalism and introduces the KKT theorem for convex problems. Linear and quadratic programming methods are covered as important application-oriented examples. Special emphasis is placed on duality, in particular, in the case of semidefinite programming. The last part of the course is devoted to deterministic and probabilistic search methods, introducing the ideas of genetic algorithms. The course provides a wide variety of examples, including applications in electronics, decision-making, machine learning, and optimal control.

Intended Learning Outcomes

By the end of this course, successful students will be able to

1. apply classical search techniques;
2. apply and understand the Lagrange formalism;
3. phrase optimization problems in terms of suitable standard types, and address them accordingly;
4. solve optimization problems by means of dedicated software packages.

Indicative Literature

S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

J. Brinkhuis & V. Tikhomiriv, Optimization: Insights and Applications, Princeton University Press, 2005.

Usability and Relationship to other Modules

- This module builds on the first year Calc/LA modules and prepares the students for more challenging optimization aspects, which will be relevant in many third year projects, particularly in the fields of machine learning, robotics, control, and communication.

Examination Type: Module Examination

Type: Written examination	Duration: 120 min Weight: 100%

Scope: Intended Learning Outcomes 1–3

Intended Learning Outcome 4 will be assessed through non graded tasks during the lecture.

Completion: To pass this module, the examination has to be passed with at least 45%.

7.20 Distributed Algorithms

Module Name Distributed Algorithms	**Module Code** CA-S-CS-803	**Level (type)** Year 3 (Specialization)	**CP** 5

Module Components

Number	Name	Type	CP
CA-CS-803	Distributed Algorithms	Lecture	5

Module Coordinator Dr. Kinga Lipskoch	**Program Affiliation** • Computer Science (CS)	**Mandatory Status** Mandatory elective for CS, SDT and RIS

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	**Frequency**	**Forms of Learning and Teaching**
☒ Algorithms and Data Structures or Core Algorithms and Data Structures	☒ None		Annually (Fall or Spring)	• Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

None

Content and Educational Aims

Distributed algorithms are the foundation of modern distributed computing systems. They are characterized by a lack of knowledge of a global state, a lack of knowledge of a global time, and inherent non-determinism in their execution. The course introduces basic distributed algorithms using an abstract formal model, which is centered on the notion of a transition system. The topics covered are logical clocks, distributed snapshots, mutual exclusion algorithms, wave algorithms, election algorithms, reliable broadcast algorithms, and distributed consensus algorithms. Process algebras are introduced as another formalism to describe distributed and concurrent systems.

The distributed algorithms introduced in this module form the foundation of computing systems that have to be scalable and fault-tolerant, e.g., large-scale distributed non-standard databases or distributed file systems. The course is recommended for students interested in the design of scalable distributed computing systems.

Intended Learning Outcomes

By the end of this module, students will be able to

1. describe and analyze distributed algorithms using formal methods such as transition systems;
2. explain different algorithms to solve election problems;
3. illustrate the limitations of time to order events and how logical clocks and vector clocks overcome these limitations;
4. apply distributed algorithms to produce consistent snapshots of distributed computations;
5. describe the differences among wave algorithms for different topologies;
6. analyze and implement distributed consensus algorithms such as Paxos and Raft;
7. use a process algebra such as communicating sequential processes or -calcul⊠s to model distributed algorithms.

Indicative Literature

Maarten van Steen, Andrew S. Tanenbaum: Distributed Systems, 3rd edition, Pearson Education, 2017. Nancy A. Lynch: Distributed Algorithms, Morgan Kaufmann, 1996.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.21 Computer Graphics

Module Name Computer Graphics	**Module Code** CA-S-CS-801	**Level (type)** Year 3 (Specialization)	**CP** 5

Module Components

Number	Name	Type	CP
CA-CS-801	Computer Graphics	Lecture	5

Module Coordinator Prof. Dr. Jürgen Schönwälder	**Program Affiliation** • Computer Science (CS)	**Mandatory Status** Mandatory elective for CS and RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall)	• Class attendance (35 hours) • Private study (70 hours) • Exam preparation (20 hours)
☒ Algorithms and Data Structures or Core Algorithms and Data Structures	☒ None		**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

None

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students will be able to

1. construct 3D geometry representations;
2. apply 3D transformations;
3. understand the algorithms and optimizations applied by graphics rendering systems;
4. explain the stages of modern computer graphics programmable pipelines
5. implement simple computer graphics applications using graphics frameworks such as OpenGL;
6. 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.

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.

Examination Type: Module Examination

Assessment Type: Written examination — Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.22 Software Engineering

Module Name Software Engineering	**Module Code** CO-561	**Level (type)** Year 2 (CORE)	**CP** 7.5

Module Component

Number	Name	Type	CP
CO-561-A	Software Engineering	Lecture	2.5
CO-561-B	Software Engineering Project	Project	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr. Peter Baumann	• Computer Science (CS)	Mandatory for CS and minor CS Mandatory elective for RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Class attendance (35 hours) • Independent study (10 hours) • Development work (132.5 hours) • Exam preparation (10 hours)
☒ Databases	☒ None			
			Duration	**Workload**
			1 semester	187.5 hours

Recommendations for Preparation

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.

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of this module, students will be able to

1. understand and apply object-oriented design patterns;
2. read and write UML diagrams;
3. contrast the benefits and drawbacks of different software development models;
4. design and plan a larger software project involving a team development effort;
5. translate requirements formulated by a customer into computer science terminology;
6. evaluate the applicability of different software engineering models for a given software development project;
7. assess the quality of a software design and its implementation;
8. apply tools that assist in the various stages of a software development process;
9. work effectively in a team toward the goals of the team.

Indicative Literature

Ian Sommerville: Software Engineering, Pearson, 2010.

Roger Pressman: Software Engineering – a Practitioner's Approach, McGraw-Hill, 2014.

Usability and Relationship to other Modules

Examination Type: Module Component Examinations Module

Component 1: Lecture

Assessment Type: Written examination Duration: 60 min
Weight: 33%

Scope: The first three intended learning outcomes of the module (the lecture module component)

Module Component 2: Project

Assessment Type: Project Assessment

Weight: 66% Scope:

The remaining intended learning outcomes of the module (the project module component)

Completion: To pass this module, the examination of each module component has to be passed with at least 45%.

7.23 Databases

Module Name	**Module Code**	**Level (type)**	**CP**
Databases	CO-560	Year 2 (CORE)	7.5

Module Components

Number	Name	Type	CP
CO-560-A	Databases	Lecture	5
CO-560-B	Databases - Project	Project	2.5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr. Peter Baumann	• Computer Science (CS)	Mandatory for CS and minor CS Mandatory elective for RIS

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills
☒ Algorithms and Data Structures	☒ None	

Frequency	**Forms of Learning and Teaching**
Annually (Fall)	• Class attendance (35 hours) • Project (97.5 hours) • Independent Studies (35 hours) • Exam preparation (20 hours)

Duration	**Workload**
1 semester	187.5 hours

Recommendations for Preparation

Working knowledge of basic data structures, such as trees, is required as well as familiarity with an object-oriented programming language. Basic knowledge of algebra is useful. For the project work, students benefit from having basic hands-on skills using Linux (the server platform in the project).

Content and Educational Aims

This module offers an introduction to databases, with emphasis on practically applicable knowledge and skills. The course starts with conceptual database design using the Entity Relationship (ER) model, followed by the relational model and SQL for querying relations. On that occasion, structures for storing relations on disk are inspected. After that, tuning opportunities are discussed, including Normal Forms, indexing, transaction management, and views, and finally – based on a brief look at Relational Algebra – query processing and optimization in the server. As today databases often are used for Web services an excursion is made to inspect the server side of Web request processing in the context of databases. This in turn prompts security considerations in databases. Concluding the relational part, the travel leads into NoSQL and NewSQL world. This widens the perspective towards data models beyond tables and redefined transaction concepts. Towards the semester end, OLAP datacubes are introduced as a practically important database application with special needs, concepts, and technology.

A hands-on group project complements the theoretical aspects: on a self-chosen topic, teams of 3 – 4 students implement the core of a web-accessible information system using python (or PHP), MariaDB, and Linux, in a guided sequence of homework assignments.

Intended Learning Outcomes

By the end of this module, students will be able to

1. read and write ER diagrams;
2. design and normalize schemas for relational databases;
3. write SQL queries and understand their evaluation in a database server;
4. know common tuning methods in relational databases;
5. explain the concept of transactions and how to use transactions in application design;
6. explain core security and privacy issues in the context of databases;
7. describe differences of selected NoSQL data models and make a requirement-driven choice;
8. describe the concept of datacubes and how databases can support it;
9. develop database-backed Web-enabled information systems, considering security aspects.

Indicative Literature

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom: Database Systems: The Complete Book. 2nd edition, Pearson, 2008.
- Elvis C. Foster, Shripad V. Godbole: Database Systems. O'Reilly, 2014
- Miguel Grinberg: Flask Web Development: Developing Web Applications with Python. O'Reilly, 2018
- Jon Duckett: PHP & MySQL: Server-side Web Development. Wiley, 2022

Usability and Relationship to other Modules

Databases form an indispensable part of today's information-hungry society, and given the emphasis on practical aspects, there is a high usability in all sectors. Among others, students can apply their knowledge in the Software Engineering module. This module serves as a default advanced level minor module.

Examination Type: Module Component Examinations

Module Component 1: Lecture

Assessment Type: Written examination Duration: 120 min

Weight: 67%

Scope: Intended learning outcomes #1 - #8

Module Component 2: Project

Assessment Type: Project assessment

Weight: 33%

Scope: Intended learning outcome #9

Completion: To pass this module, the examination of each module component has to be passed with at least 45%.

7.24 Digital Design

Module Name Digital Design	**Module Code** CA-S-ECE-803	**Level (type)** Year 3 (Specialization)	**CP** 5

Module Components

Number	Name	Type	CP
CA-ECE-803	Digital Design	Lecture/Lab	5

Module Coordinator Dr. Fangning Hu	**Program Affiliation** • Electrical and Computer Engineering (ECE)	**Mandatory Status** Mandatory elective for ECE, RIS and CS

Entry Requirements Pre-requisites ☒ None	Co-requisites ☒ None	Knowledge, Abilities, or Skills	**Frequency** Annually (Fall)	**Forms of Learning and Teaching** • Lecture/Lab (35 hours) • Private study (90 hours)
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

Students may prepare themselves with books like “Brent E. Nelson, Designing Digital Systems, 2005” and “Pong P. Chu, RTL Hardware Design Using VHDL, A John Wiley & Sons, Inc, Publication, 2006”

Content and Educational Aims

The current trend of digital system design is towards hardware description languages (HDLs) that allow compact description of very complex hardware constructs. The module provides a sound introduction to basic components of a digital system such as logic gates, multiplexers, decoders, flip-flops and registers as well as VHDLs such as types, signals, sequential and concurrent statements. Methods and principle of designing complex digital systems such as finite state machines, hierarchical design, pipelined design, RTL design methodology and parameterized design will also be introduced. Students will learn VHDL for programming FPGA boards to realize small digital systems in hardware (i.e. on FPGA boards). Such digital systems could be adders, multiplexers, control units, multipliers, asynchronous serial communication modules (UART). At the end of the module, the students should be able to design a simple digital system by VHDL on an FPGA board.

Intended Learning Outcomes

By the end of this module, students will be able to

1. understand the principle of digital system design based on standard building blocks and components;
2. design a complex digital system;
3. understand the limitations of a given hardware platform (here FPGAs), modify algorithms where necessary, and structure them suitably in order to optimize performance and complexity;
4. use a typical development system;
5. program in VHDL;
6. program an FPGA board.

Indicative Literature

Brent E. Nelson, Designing Digital Systems with SystemVerilog, 2018, ISBN-13: 978-1980926290 Pong P. Chu,

RTL Hardware Design Using VHDL, Wiley-IEEE Press, 2006, ISBN-13: 978-0471720928

Usability and Relationship to other Modules

- This module introduces how to design digital systems and how to realize them on a FPGA board which could also serve as a specialization module for students from Computer Science and Robotics and Intelligent Systems.

Examination Type: Module Examination

Assessment Type: Written examination	Duration: 120 min
Scope: All intended learning outcomes of the module	Weight: 100%

Completion: To pass this module, the examination has to be passed with at least 45%.

7.25 PCB Design and Measurement Automation

Module Name PCB Design and Measurement Automation	**Module Code** CO-527	**Level (type)** Year 2 (CORE)	**CP** 5

Module Components

Number	Name	Type	CP
CO-527-A	PCB Design and Measurement Automation	Lab	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr.-Ing. Werner Henkel	• Electrical and Computer Engineering (ECE)	Mandatory for ECE Mandatory elective for RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Lab (59.5 hours) • Private Study (65.5 hours)
☒ General Electrical Engineering I ☒ General Electrical Engineering II OR Mathematical and Physical Foundations of Robotics I & II	☒ None	• Knowledge of Fourier series and transforms • Basic knowledge of electronics components and circuits • Matlab	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

Download material from corresponding Web pages and get to know the tasks and how the tools and equipment works.

Content and Educational Aims

The module (lab) covers mainly two aspects that are seen to be important for employability. One share of the lab deals with measurement automation. Similar tasks, one also finds in industrial automation or monitoring, sometimes using the same tools. Students will learn to use Matlab and Labview for measurement automation tasks. In there, students will also get acquainted with more advanced measurement equipment, like high-end digital scopes, network, and spectrum analyzers. The students will measure standard telephone cables in their properties, which will require a treatment of transmission line theory and transformers/baluns. These theoretical aspects will also be covered.

The second major aspect handled in the lab makes students aware that electrical/electronic components have non-ideal behaviors, e.g., that a capacitor can act as an inductor in some frequency range. It makes students also aware of the problems in selecting the right component for a certain function inside a circuit, caring not just for the frequency range and the variation of properties with frequency, but also power, current, and voltage limits. Then, a typical circuit design path will be taught, starting from schematics to placement of components and routing. Important aspects of printed circuit board design are treated, like how analog and digital power supplies have to be realized, how mass connections should look like, what measures have to be taken to block unwanted signal coupling is avoided, e.g., blocking capacitors, star-like power supply wiring.

Students also practice scientific writing in line with scientific writing rules as a preparation for their BSc thesis.

Intended Learning Outcomes

By the end of this module, students should be able to

1. use vector network analyzers, spectrum analyzers, and more advanced digital scopes;
2. learn how to program with LabVIEW;
3. remotely control measurement equipment using Matlab or LabVIEW;
4. describe principles of remote control;
5. know transmission line theory and how transformers/baluns are modeled;
6. measure and determine line parameters;
7. taking non-ideal behavior of passive and active components into account and be able to select components according to their parameters and limitations;
8. design printed circuit boards (PCB) with typical tools and a typical design cycle consisting of schematics, placement, and routing;
9. design analog and digital power routes, shielding ground connections, use measures to block unwanted ingress and coupling;
10. organize work contributions of group members in the lab and in reporting;
11. write reports in line with scientific writing rules as a preparation for their BSc thesis.

Usability and Relationship to other Modules

- This module builds on previous electronics knowledge and rounds this knowledge up with the final PCB design.
- Having learned to use Matlab in earlier modules, mostly for signal processing tasks, this module shows another application and provides a view into graphical programming as another option which they have seen earlier in the form of Simulink
- The module prepares students for a thesis with PCB design aspects.
- Serves as a mandatory elective 3rd year Specialization module for RIS major students.

Indicative Literature

Hank Zumbahlen Ed., Basic Linear Design, Analog Devices, 2007. Walt Jung

Ed., Op Amp Applications, Analog Devices, 2005.

Tim Williams, The Circuit Designer's Companion, 3rd ed., Newnes, 2012.

National Instruments, LabVIEW, Getting Started with LabVIEW, 2007.

Examination Type: Module Examination

Assessment Component 1: Written examination
Duration: 120 min
Weight: 50%

Scope: Intended learning outcomes of the lecture/theory component (4, 5, 7, 9).

Assessment Component 2: Laboratory reports
Length: 5-10 pages per experiment session
Weight: 50%

Scope: Intended learning outcomes of the lab (1-3, 6-11).

Completion: This module is passed with an assessment-component weighted average grade of 45% or higher.

7.26 Information Theory

Module Name	**Module Code**	**Level (type)**	**CP**
Information Theory	CO-525	Year 2 (CORE)	5

Module Components

Number	Name	Type	CP
CO-525-A	Information Theory	Lecture	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr.-Ing. Werner Henkel	• Electrical and Computer Engineering (ECE)	Mandatory for ECE Mandatory elective for CS, PHDS and RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Lectures (35 hours) • Private Study (90 hours)
			Duration	**Workload**
☒ None	☒ None	• Signals and Systems contents, such as DFT and convolution • Notion of probability, combinatorics basics as taught in Methods module “Probability and Random Processes"	1 semester	125 hours

Recommendations for Preparation

Some basic knowledge of communications and sound understanding of probability is recommended. Hence, it is strongly advised to take the methods and skills course Probability and Random Processes prior to this module. Nevertheless, probability basics will also be revised within the module.

Content and Educational Aims

Information theory serves as the most important foundation for communication systems. The module provides an analytical framework for modeling and evaluating point-to-point and multi-point communication. After a short rehearsal of probability and random variables and some excursion to random number generation, the key concept of information content of a signal source and information capacity of a transmission medium are precisely defined, and their relationships to data compression algorithms and error control codes are examined in detail. The module aims to install an appreciation for the fundamental capabilities and limitations of information transmission schemes and to provide the mathematical tools for applying these ideas to a broad class of communications systems.
The module contains also a coverage of different source-coding algorithms like Huffman, Lempel-Ziv-(Welch), Shannon-Fano-Elias, Arithmetic Coding, Runlength Encoding, Move-to-Front transform, PPM, and Context Tree Weighting. In Channel coding, finite fields, some basic block and convolutional codes, and the concept of iterative decoding will be introduced. Aside from source and channel aspects, an introduction to security is given, including public-key cryptography. Information theory is a standard module in every communications-oriented Bachelor’s
program.

Intended Learning Outcomes

By the end of this module, students should be able to

1. explain what is understood as the information content of data and the corresponding limits of data compression algorithms;
2. design and apply fundamental algorithms in data compression;
3. explain the information theoretic limits of data transmission;
4. apply the mathematical basics of channel coding and cryptography;
5. implement some channel coding schemes;
6. differentiate the principles of encryption and authentication schemes and implement discussed procedures.

Indicative Literature

Thomas M. Cover, Joy A. Thomas, Elements of Information Theory, 2nd ed., Wiley, Sept. 2006. David

Salomon, Data Compression, The Complete Reference, 4th ed., Springer, 2007.

Usability and Relationship to other Modules

- Although not a mandatory prerequisite, this module is ideally taken before Coding Theory (CA-ECE-802)
- All communications-related modules are naturally based on information theory
- Students from Computer Science or related programs, also students taking Bio-informatics modules, profit from information-theoretic knowledge and source coding (compression) algorithms. Students from Computer Science would also be interested in the algebraic basics for error-correcting codes and cryptology, fields which area also introduced shortly.
- Serves as a mandatory elective 3rd year Specialization module for CS and RIS major students.

Examination Type: Module Examination

Assessment Type: Written examination Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%.

7.27 Stochastic Processes

Module Name	**Module Code**	**Level (type)**	**CP**
Stochastic Processes	CA-S-MMDA-805	Year 2/3 (Specialization)	5

Module Components

Number	Name	Type	CP
CA-MMDA-805	Stochastic Processes	Lecture	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Dr. Keivan Mallahi Karai	• Mathematics, Modeling and Data Analytics (MMDA)	Mandatory elective for MMDA, and RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Biennially (Spring)	• Lectures (35 hours) • Private study (90 hours)
☒ Matrix Algebra and Advanced Calculus II and Probability and Random Processes	☒None	• None beyond formal pre-requisites	**Duration**	**Workload**
			1 semester	125 hours

Recommendations for Preparation

Review of Probability and Analysis

Content and Educational Aims

This module serves as an introduction to the theory of stochastic processes. It starts with a review of Kolmogorov axioms for probability spaces and continues by providing a rigorous treatment of topics such as the independence of events and Borel-Cantelli Lemma, Kolmogorov's zero-one law, random variables, expected value and variance, the weak and strong laws of large numbers, and the Central limit theorem. More advanced topics that will follow include finite and countable state Markov chains, Galton-Watson trees, and the Wiener process. Several relevant applications that will be discussed are percolation on graphs, the application of Markov chains to sampling problems, and probabilistic methods in graph theory. The module also includes examples from mathematical finance.

Intended Learning Outcomes

By the end of the module, students will be able to

1. demonstrate their mastery of basic stochastic methods;
2. develop ability to use stochastic processes to model real-world problems, e.g. in finance;
3. analyze the definition of basic probabilistic objects, and their numerical features;
4. formulate and design methods and algorithms for solving applied problems based on ideas from stochastic processes.

Indicative Literature

R. Durrette (2019). Probability: Theory and Examples. Cambridge: Cambridge University Press.

A. Koralov and Ya. Sinai (2007). Theory of Probability and Random Processes, Berlin: Springer.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.28 Stochastic Modeling and Financial Mathematics

Module Name	Module Code	Level (type)	CP
Stochastic Modeling and Financial Mathematics	CA-S-MMDA-803	Year 2 and 3 (Specialization)	5

Module Components

Number	Name	Type	CP
CA-MMDA-803	Stochastic Modeling and Financial Mathematics	Lecture	5

Module Coordinator	Program Affiliation	Mandatory Status
Prof. Dr. Sören Petrat	• Mathematics, Modeling, and Data Analytics (MMDA)	Mandatory elective for MMDA, PHDS, SDT and RIS

Entry Requirements			Frequency	Forms of Learning and Teaching
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring/Fall)	• Lectures (35 hours) • Private Study (90 hours)
☒ Matrix Algebra and Advanced Calculus I & II	☒ none	Good command of Calculus, Linear Algebra, and basic probability basic Python programming	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

- Review the content of Matrix Algebra & Advanced Calculus II
- Review Python programming
- Pre-install Anaconda Python on your own laptop and know how to edit and start simple Python programs in a Python IDE like Spyder (which comes bundled as part of Anaconda Python).

Content and Educational Aims

This module is a first hands-on introduction to stochastic modeling. Examples will mostly come from the area of Financial Mathematics, so that this module plays a central role in the education of students interested in Quantitative Finance and Mathematical Economics. The module is taught as an integrated lecture-lab, where short theoretical units are interspersed with interactive computation and computer experiments.

Topics include a short introduction to the basic notions of financial mathematics, binomial tree models, discrete Brownian paths, stochastic integrals and ODEs, Ito's Lemma, Monte-Carlo methods, finite differences solutions, the Black-Scholes equation, and an introduction to time series analysis, parameter estimation, and calibration. Towards the end, the Fokker-Planck equation, Ornstein-Uhlenbeck processes, and nonlinear Stochastic Partial Differential Equations are discussed, and connections to applications in physics and other areas of mathematics are made. Students will program and explore all basic techniques in a numerical programming environment and apply these algorithms to real data whenever possible.

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. apply fundamental concepts of deterministic and stochastic modeling;
2. design, conduct, and interpret controlled in-silico scientific experiments;
3. analyze the basic concepts of financial mathematics and their role in finance;
4. write computer code for basic financial calculations, binomial trees, stochastic differential equations, stochastic integrals and time series analysis;
5. compare their programs and predictions in the context of real data;
6. demonstrate the usage of a version control system for collaboration and the submission of code and reports.

Indicative Literature

- Y.-D. Lyuu (2002). Financial Engineering and Computation - Principles, Mathematics, Algorithms. Cambridge: Cambridge University Press.

- J.C. Hull (2015). Options, Futures and other Derivatives, 9th edition. New York: Pearson.
- A. Etheridge (2002). A Course in Financial Calculus. Cambridge: Cambridge University Press.
- D.J. Higham (2001). An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations, SIAM Rev. 43(3):525-546.
- D.J. Higham (2004). Black-Scholes Option Valuation for Scientific Computing Students, Computing in Science & Engineering 6(6):72-79.

Usability and Relationship to other Modules

- This module is part of the core education in Mathematics, Modeling and Data Analytics.
- It is also valuable for students in Physics and Data Science, Computer Science, Data Engineering, RIS, and ECE, either as part of a minor in Mathematics, or as an elective module.

Examination Type: Module Examination

Assessment Type: Portfolio Assessment Weight: 100%

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examination has to be passed with at least 45%.

7.29 Operations Research

Module Name Operations Research	**Module Code** CO-583	**Level (type)** Year 2 (CORE)	**CP** 5

Module Components

Number	Name	Type	CP
CO-583-A	Operations Research	Lecture	5

Module Coordinator Dr. Stanislav Chankov	**Program Affiliation** • Industrial Engineering & Management (IEM)	**Mandatory Status** Mandatory for IEM Mandatory elective for RIS

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	**Frequency**	**Forms of Learning and Teaching**
☒ None	☒ None	• Basic spreadsheet software skills (e.g. MS Excel) • basic calculus and matrix algebra • basic knowledge in logistics	Annually (Fall) **Duration** 1 semester	• Lectures (35 hours) • Private Study (90 hours) **Workload** 125 hours

Recommendations for Preparation

Revise basic calculus, matrix algebra and spreadsheet software functions.

Content and Educational Aims

Operations research is an interdisciplinary mathematical science that focuses on the effective use of technology by organizations. By employing techniques such as mathematical modeling, statistical analysis, and mathematical optimization, operations research finds optimal or near-optimal solutions to complex decision-making problems. Operations Research is concerned with determining the maximum (of profit, performance, or yield) or the minimum (of loss, risk, or cost) of some real-world objective. This module introduces students to the modelling of decision problems and the use of quantitative methods and techniques for effective decision-making.

Intended Learning Outcomes

By the end of this module, students will be able to

1. calculate optimal or near-optimal solutions to complex decision-making problems using operations research methods;
2. design mathematical models for business problems;
3. apply techniques such as linear programming, dynamic programming or stochastic programming to solve business problems;
4. resolve common network optimization problems such as transportation, shortest path, minimum spanning tree, and maximum flow problems.

Indicative Literature

Hillier, F. S. & Lieberman, G.J. (2009). Introduction to Operations Research. McGraw-Hill. New York, NY.

Usability and Relationship to other Modules

- Serves as a 3rd year Specialization module for major students in RIS

Examination Type: Module Examination

Assessment Type: Written examination Duration: 120 minutes
Weight: 100 %

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%.

7.30 Web Application Development

Module Name Web Application Development	**Module Code** CA-S-CS-804	**Level (type)** Year 3 (Specialization)	**CP** 5

Module Components

Number	Name	Type	CP
CA-CS-804-A	Web Application Development	Lecture	2.5
CA-CS-804-B	Web Application Development - Project	Project	2.5

Module Coordinator Prof. Dr. Jürgen Schönwälder	**Program Affiliation** • Computer Science (CS)	**Mandatory Status** Mandatory elective for CS and RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites ☒ Databases and Web Services	Co-requisites ☒ None	Knowledge, Abilities, or Skills	Annually (Spring)	• Class attendance (17.5 hours) • Private study (40 hours) • Project work (50 hours) • Exam preparation (17.5 hours)
			Duration 1 semester	**Workload** 125 hours

Recommendations for Preparation

None

Content and Educational Aims

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

By the end of this module, students will be able to

1. explain the document object model behind HTML and its relation to CSS;
2. discuss the principles and basic mechanisms of reactive website design;
3. analyze the interactions between web applications and web services.

4. use languages such as Java, Kotlin, or Swift to implement mobile web applications;
5. 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.

Usability and Relationship to other Modules

Examination Type: Module Component Examinations Module

Component 1: Lecture

Assessment Type: Written examination — Duration: 120 min, Weight: 50%

Scope: First group of intended learning outcomes of the module

Module Component 2: Project

Assessment Type: Project Assessment — Weight: 50%

Scope: Second group of intended learning outcomes of the module

Completion: To pass this module, the examination of each module component has to be passed with at least 45%.

7.31 Parallel and Distributed Computing

Module Name Parallel and Distributed Computing	**Module Code** MDE-CS-02	**Level (type)** Year 2 (Elective)	**CP** 5

Module Components

Number	Name	Type	CP
MDE-CS-02	Parallel and Distributed Computing	Lecture	5

Module Coordinator Prof. Dr. Stefan Kettemann	**Program Affiliation** ▪ MSc Data Engineering (DE)	**Mandatory Status** Mandatory elective for DE, CSSE (MSc), RIS (BSc) and CS (BSc)

Entry Requirements			**Frequency** Annually (Fall)	**Forms of Learning and Teaching** ▪ Lecture (35 hours) ▪ Private study (90 hours)
Pre-requisites	Co-requisites ☒ None	Knowledge, Abilities, or Skills ▪ Basic knowledge in C/C++ ▪ Mandatory proficiency in Python	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

If no knowledge in C/C++ is present, interested students are encouraged get a basic understanding of C/C++ (via online material) in order to better understand some of the discussed concepts.

Content and Educational Aims

In the recent years, the development of parallel and cloud computing has opened the door for Big Data analysis and processing. This module aims at providing an overview and introduction to the vast field of parallel and cloud computing. In traditional parallel computing, we aim to develop notions for different parallelization models (shared-memory,distributed-memory, SIMD, SIMT), get to know appropriate programming methodologies for high performance dataanalysis (OpenMP / MPI) and aim at understanding performance and scalability in this field (weak vs. strong scaling, Amdahl's law).This fundamental knowledge will then be carried over to recent developments in cloud computing, where distributed processing frameworks (Spark / Hadoop MapReduce / Dask), based on appropriated deployment infrastructures, are in the process to become De Facto standards for Big Data processing and analysis. We will approach these technologies from a practical point of view and aim at developing the necessary knowledge to carry out scalable machine learning and data processing on Big Data.

Intended Learning Outcomes

By the end of this module, students should be able to

1. understand theory and fundamentals of parallelization models (shared-/distributed memory, SIMD, SIMT)
2. explain and apply parallel programming methodologies (OpenMP / MPI)
3. describe and analyze performance and scalability (weak vs. strong scaling, ...)
4. Understand basic principles of distributed and cloud computing
5. use distributed processing frameworks (Spark / Hadoop MapReduce / Dask) for scalable distributed calculations
6. develop scalable machine learning and data processing on Big Data

Indicative Literature

Zaccone, Python Parallel Programming Cookbook, O'Reilly.

J.C. Daniel, Data Science with Python and Dask, Manning Publications.

Z. Radtka, D. Miner, Hadoop with Python. Hadoop with Python, O'Reilly.
Usability and Relationship to other Modules
Examination Type: Module Examination Assessment Type: Written Exam — Duration: 120 minutes Weight: 100% Scope: All intended learning outcomes of this module. Completion: To pass this module, the examination has to be passed with at least 45%.

7.32 Internship / Startup and Career Skills

Module Name Internship / Startup and Career Skills	**Module Code** CA-INT-900	**Level (type)** Year 3 (CAREER)	**CP** 15
Module Components			
Number	Name	Type	CP
CA-INT-900-0	Internship	Internship	15

Module Coordinator Clémentine Senicourt & Dr. Tanja Woebs (SCS Organization); SPC / Faculty Startup Coordinator (Academic responsibility)	**Program Affiliation** • CAREER module for undergraduate study programs	**Mandatory Status** Mandatory for all undergraduate study programs except IEM

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring/Fall)	• Internship/Start-up • Internship event • Seminars, info-sessions, workshops and career events • Self-study, readings, online tutorials
☒ at least 15 CP from CORE modules in the major	☒ None	• Information provided on CSC pages (see below) • Major specific knowledge and skills	**Duration** 1 semester	**Workload** 375 Hours consisting of: • Internship (308 hours) • Workshops (33 hours) • Internship Event (2 hours) • Self-study (32 hours)

Recommendations for Preparation

- 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

Content and Educational Aims

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 Student Career Support (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

Intended Learning Outcomes

By the end of this module, students should be able to

1. describe the scope and the functions of the employment market and personal career development;
2. 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 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 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 professional decisions based on theoretical knowledge and academic methods;
6. reflect on their professional conduct in the context of the expectations of and consequences for employers and their society;
7. reflect on and set their own targets for the further development of their knowledge, skills, interests, and values;
8. 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 observations and reflections in a professional network.

Indicative Literature

Not specified

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.

Examination Type: Module Examination

Assessment Type: Internship Report or Business Plan and Reflection	Length: approx. 3.500 words
Scope: All intended learning outcomes	Weight: 100%

7.33 Bachelor Thesis and Seminar

Module Name Bachelor Thesis and Seminar RIS			**Module Code** CA-RIS-800	**Level (type)** Year 3 (CAREER)	**CP** 15
Module Components					
Number	Name			Type	CP
CA-RIS-800-T	Thesis RIS			Thesis	12
CA-RIS-800-S	Thesis Seminar RIS			Seminar	3
Module Coordinator Study Program Chair	**Program Affiliation** • All undergraduate programs			**Mandatory Status** Mandatory for all undergraduate programs	
Entry Requirements			**Frequency** Annually (Spring)	**Forms of Learning and Teaching** • Self-study/lab work (350 hours) • Seminars (25 hours)	
Pre-requisites ☒ 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.	Co-requisites ☒ None	Knowledge, Abilities, or Skills • 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.	**Duration** 14-week lecture period	**Workload** 375 hours	

Recommendations for Preparation

- Identify an area or a topic of interest and discuss this with your prospective supervisor in a timely manner.
- 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 the University's Code of Academic Integrity and Guidelines to Ensure Good Academic Practice.

Content and Educational Aims

This module is a mandatory graduation requirement for all undergraduate students to demonstrate their ability to address a problem from their respective major subject independently using academic/scientific methods within a set time frame. Although supervised, this module requires students to be able to work independently and systematically and set their own goals in exchange for the opportunity to explore a topic that excites and interests them personally and that a faculty member is interested in supervising. Within this module, students apply their acquired knowledge about their major discipline and their learned skills and methods for conducting 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 research results.

This module consists of two components, an independent thesis and an accompanying seminar. The thesis component 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 a conclusion. The seminar provides students with the opportunity to practice their ability to present, discuss, and justify their and other students' approaches, methods, and results at various stages of their research in order to improve their academic writing, receive and reflect on formative feedback, and therefore grow personally and professionally.

Intended Learning Outcomes

On completion of this module, students should be able to

1. independently plan and organize advanced learning processes;
2. design and implement appropriate research methods, taking full account of the range of alternative techniques and approaches;
3. collect, assess, and interpret relevant information;
4. draw scientifically-founded conclusions that consider social, scientific, and ethical factors;
5. apply their knowledge and understanding to a context of their choice;
6. develop, formulate, and advance solutions to problems and debates within their subject area, and defend these through argument;
7. discuss information, ideas, problems, and solutions with specialists and non-specialists.

Indicative Literature

Justin Zobel, Writing for Computer Science, 3rd edition, Springer, 2015.

Usability and Relationship to other Modules

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

Examination Type: Module Component Examinations

Module Component 1: Thesis
Assessment type: Thesis
Scope: All intended learning outcomes, mainly 1-6.

Length: approx. 10,000 – 14,000 words (25–35 pages), excluding front and back matter.
Weight: 80%

Module Component 2: Seminar
Assessment type: Presentation

Duration: approx. 15 to 30 minutes
Weight: 20%

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

Completion: To pass this module, both module component examinations have to be passed with at least 45%.

Two separate assessments are justified by the size of this module and the fact that the justification of solutions to problems and arguments (ILO 6) and discussion (ILO 7) should at least have verbal elements. The weights of the types of assessments are commensurate with the sizes of the respective module components.

8 Constructor Track Modules

8.1 Methods Modules

8.1.1. Elements of Linear Algebra

Module Name Elements of Linear Algebra			**Module Code** CTMS-MAT-24	**Level (type)** Year 1 (Methods)	**CP** 5
Module Components					
Number	Name			Type	CP
CTMS-24	Elements of Linear Algebra			Lecture	5
Module Coordinator Dr. Keivan Mallahi Karai	**Program Affiliation** • CONSTRUCTOR Track Area			**Mandatory Status** Mandatory elective for CS, RIS and SDT	
Entry Requirements			**Frequency** Annually (Fall)	**Forms of Learning and Teaching** • Lectures (35 hours) • Private study (90 hours)	
Pre-requisites ☒ None	Co-requisites ☒ None	Knowledge, Abilities, or Skills 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)	**Duration** 1 semester	**Workload** 125 hours	

Recommendations for Preparation

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

Content and Educational Aims

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

Intended Learning Outcomes
By the end of the module, students will be able to

1. apply 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 the mathematical structures in an unfamiliar context and translate them into a mathematical problem statement;
3. 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 prove results which are direct consequences of those proved in the lectures;
5. 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)

Usability and Relationship to other Modules

- A rigorous treatment of this topic is provided in the module “Linear Algebra.”

Examination Type: Module Examination

Assessment type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examination has to be passed with at least 45%.

8.1.2 Elements of Calculus

Module Name	Module Code	Level (type)	CP
Elements of Calculus	CTMS-MAT-25	Year 1 (Methods)	5

Module Components

Number	Name	Type	CP
CTMS-25	Elements of Calculus	Lecture	5

Module Coordinator	Program Affiliation	Mandatory Status
Dr. Keivan Mallahi Karai	• CONSTRUCTOR Track Area	Mandatory elective for CS, RIS and SDT

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Lectures (35 hours) • Private study (90 hours)
☒	☒ None	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.	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

Review the content of Linear Algebra

Content and Educational Aims

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

Intended Learning Outcomes
By the end of the module, students will be able to

1. apply 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 the mathematical structures in an unfamiliar context and translate them into a mathematical problem statement;
3. 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 prove results which are direct consequences of those proved in the lectures;
5. 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)

Usability and Relationship to other Modules

- A rigorous treatment of this topic is provided in the module “Analysis”.

Examination Type: Module Examination

Assessment type: Written examination Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examination has to be passed with at least 45%.

8.1.3 Matrix Algebra and Advanced Calculus I

Module Name Matrix Algebra and Advanced Calculus I			**Module Code** CTMS-MAT-22	**Level (type)** Year 1 (Methods)	**CP** 5
Module Components					
Number	Name			Type	CP
CTMS-22	Matrix Algebra and Advanced Calculus I			Lecture	5
Module Coordinator Dr. Keivan Mallahi-Karai	**Program Affiliation** • CONSTRUCTOR Track Area			**Mandatory Status** Mandatory for ECE and SDT MMDA, PHDS. Mandatory elective for CS, and RIS	
Entry Requirements			**Frequency**	**Forms of Learning and Teaching**	
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring/Fall)		
			Duration	**125 hours**	
☒ none	☒ none	• Knowledge of pre-calculus ideas (sets and functions, elementary functions, polynomials) and analytic geometry (equations of lines, systems of linear equations, dot product, polar coordinates) at High School level. Familiarity with ideas of calculus is helpful.	1 semester		
Recommendations for Preparation Review of high school mathematics.					

Content and Educational Aims

This module is the first in a sequence including advanced mathematical methods at the university level at a level higher than the course Calculus and Linear Algebra I. The course comprises the following topics:

- Number systems, complex numbers
- The concept of function, composition of functions, inverse functions
- Basic ideas of calculus: Archimedes to Newton
- The notion of limit for functions and sequences and series
- Continuous function and their basic properties
- Derivatives: rate of change, velocity and applications
- Mean value theorem and estimation, maxima and minima, convex functions
- Integration, change of variables, Fundamental Theorem of Calculus
- Applications of the integral: work, area, average value, centre of mass
- Improper Integrals, Mean value theorem for integrals
- Taylor series
- Ordinary differential equations, examples, solving first order linear differential equations
- Basic ideas of numerical analysis, Newton's method, asymptotic formulas
- Review of elementary analytic geometry, lines, conics
- Vector spaces, linear independence, bases, coordinates
- Linear maps, matrices and their algebra, matrix inverses
- Gaussian elimination, solution space
- Determinants

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. apply the methods described in the content section of this module description to the extent that they can
2. solve standard text-book problems reliably and with confidence;
3. recognize the mathematical structures in an unfamiliar context and translate them into a mathematical problem statement;
4. 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

Advanced Calculus, G.B. Folland (Pearson, 2002) Linear

Algebra, S. Lang (Springer Verlag, 1986) Mathematical

Methods for Physics and Engineering,

K. Riley, M. Hobson, S. Bence (Cambridge University Press, 2006)

Usability and Relationship to other Modules

- Calculus and Linear Algebra I can be substituted with this module after consulting academic advisor
- A more advanced treatment of multi-variable Calculus, in particular, its applications in Physics and Mathematics, is provided in the second-semester module “Applied Mathematics”. All students taking “Applied Mathematics” are expected to take this module as well as the module topics are closely synchronized.
- The second-semester module “Linear Algebra” provides a complete proof-driven development of the theory of Linear Algebra. Diagonalization is covered more abstractly, with particular emphasis on degenerate cases. The Jordan normal form is also covered in “Linear Algebra”, not in this module.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min

Weight: 100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%

8.1.4 Matrix Algebra and Advanced Calculus II

Module Name Matrix Algebra and Advanced Calculus II	**Module Code** CTMS-MAT-23	**Level (type)** Year 1 (Methods)	**CP** 5

Module Components

Number	Name	Type	CP
CTMS-23	Matrix Algebra and Advanced Calculus II	Lecture	5

Module Coordinator Dr. Keivan Mallahi Karai	**Program Affiliation** • CONSTRUCTOR Track Area	**Mandatory Status** Mandatory for SDT, ECE, MMDA and PHDS Mandatory elective for CS and RIS

Entry Requirements			**Frequency** Annually (Spring)	**Forms of Learning and Teaching** • Lectures (35 hours) • Private study (90 hours)
Pre-requisites ☒ Matrix Algebra and Advanced Calculus I	Co-requisites ☒ none	Knowledge, Abilities, or Skills • None beyond formal pre-requisites	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

Review the content of Matrix Algebra and Advanced Calculus I

Content and Educational Aims

- Coordinate systems, functions of several variables, level curves, polar coordinates
- Continuity, directional derivatives, partial derivatives, chain rule (version I)
- derivative as a matrix, chain rule (version II), tangent planes and linear approximation, gradient, repeated partial derivatives
- Minima and Maxima of functions of several variables, Lagrange multipliers
- Multiple integrals, iterated integrals, integration over standard regions, change of variables formula
- Vector fields, parametric representation of curves, line integrals and arc length, conservative vector fields
- Potentials, Green's theorem in the plane
- Parametric representation of surfaces
- Vector products and normal surface integrals
- Integral theorems by Stokes and Gauss, physical interpretations
- Basics of differential forms and their calculus, connection to gradient, curl, and divergence
- Eigenvalues and eigenvectors, diagonalisable matrices
- Inner product spaces, Hermitian and unitary matrices
- Matrix factorizations: Singular value decomposition with applications, LU decomposition, QR decomposition
- Linear constant-coefficient ordinary differential equations, application to mechanical vibrations and electrical oscillations
- Periodic functions, Fourier series

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. understand the definitions of continuity, derivative of a function as a linear transformation, multivariable integrals, eigenvalues and eigenvectors and associated notions.
2. apply the methods described in the content section of this module description to the extent that they can
3. evaluate multivariable integrals using definitions or by applying Green and Stokes theorem.
4. evaluate various decompositions of matrices
5. solve standard text-book problems reliably and with confidence;

6. recognize the mathematical structures in an unfamiliar context and translate them into a mathematical problem statement;
7. 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

Advanced Calculus, G.B. Folland (Pearson, 2002) Linear

Algebra, S. Lang (Springer Verlag, 1986) Mathematical

Methods for Physics and Engineering,

K. Riley, M. Hobson, S. Bence (Cambridge University Press, 2006) Vector

Calculus, Linear Algebra, and Differential Forms: A Unified

Approach, J.H. Hubbard, B. Hubbard (Pearson, 1998)

Usability and Relationship to other Modules

- This module can substitute Calculus and Linear Algebra II after consulting academic advisor.
- Methods of this course are applied in the module Mathematical Modeling.
- The second-semester module Linear Algebra provides a more rigorous and more abstract treatment of some of the notions discussed in this module.

Examination Type: Module Examination

Assessment type: Written examination

Length/duration: (120min)
Weight: 100 %

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examination has to be passed with at least 45%

8.1.5 Probability and Random Processes

<table>
<tr><td colspan="3"><b>Module Name</b>
Probability and Random Processes</td><td><b>Module Code</b>
CTMS-MAT-12</td><td><b>Level (type)</b>
Year 2
(Methods)</td><td><b>CP</b>
5</td></tr>
<tr><td colspan="6"><b>Module Components</b></td></tr>
<tr><td>Number</td><td colspan="3">Name</td><td>Type</td><td>CP</td></tr>
<tr><td>CTMS-12</td><td colspan="3">Probability and random processes</td><td>Lecture</td><td>5</td></tr>
<tr><td><b>Module Coordinator</b>
Dr. Keivan Mallahi Karai</td><td colspan="3"><b>Program Affiliation</b>
CONSTRUCTOR Track Area</td><td colspan="2"><b>Mandatory Status</b>
Mandatory for CS, SDT, ECE, MMDA, PHDS and RIS</td></tr>
<tr><td colspan="3"><b>Entry Requirements</b></td><td rowspan="2"><b>Frequency</b>
Annually (Fall)</td><td colspan="2" rowspan="2"><b>Forms of Learning and Teaching</b>
Lectures (35 hours)
Private study (90 hours)</td></tr>
<tr><td>Pre-requisites</td><td>Co- requisites</td><td>Knowledge, Abilities, or Skills</td></tr>
<tr><td>☒ Matrix Algebra and Advanced Calculus II or Elements of Algebra and Elements of Calculus</td><td>☒ None</td><td>• 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.</td><td><b>Duration</b>
1 semester</td><td colspan="2"><b>Workload</b>
125 hours</td></tr>
<tr><td colspan="6"><b>Recommendations for Preparation</b>
Review all of the first-year calculus and linear algebra modules as indicated in “Entry Requirements – Knowledge, Ability, or Skills” above.</td></tr>
</table>

Content and Educational Aims

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.

Intended Learning Outcomes

By the end of the module, students will be able to

1. 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 the probabilistic structures in an unfamiliar context and translate them into a mathematical problem statement;
3. 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.

Usability and Relationship to other Modules

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

Examination Type: Module Examination

Assessment type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examination has to be passed with at least 45%.

8.1.6 Numerical Methods

Module Name	**Module Code**	**Level (type)**	**CP**
Numerical Methods	CTMS-MAT-13	Year 2 (Methods)	5

Module Components

Number	Name	Type	CP
CTMS-13	Numerical Methods	Lecture	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Dr. Keivan Mallahi Karai	• CONSTRUCTOR Track Area	Mandatory for ECE Mandatory elective for CS and RIS

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Spring)	• Lectures (35 hours) • Private study (90 hours)
☒ None	☒ None	• Knowledge of Calculus (functions, inverse functions, sets, real numbers, sequences and limits, polynomials, rational functions, trigonometric functions, logarithm and exponential function, parametric equations, tangent lines, graphs, derivatives, anti-derivatives, elementary techniques for solving equations) • Knowledge of Linear Algebra (vectors, matrices, lines, planes, n- dimensional Euclidean vector space, rotation, translation, dot product (scalar product), cross product, normal vector, eigenvalues, eigenvectors, elementary techniques for solving systems of linear equations)	**Duration** 1 semester	**Workload** 125 hours

Recommendations for Preparation

Taking Calculus and Elements of Linear Algebra II before taking this module is recommended, but not required. A thorough review of Calculus and Elements of Linear Algebra, with emphasis on the topics listed as “Knowledge, Abilities, or Skills” is recommended.

Content and Educational Aims

This module covers calculus-based numerical methods, in particular root finding, interpolation, approximation, numerical differentiation, numerical integration (quadrature), and a first introduction to the numerical solution of differential equations.

The lecture comprises the following topics

- number representations
- Gaussian elimination
- LU decomposition
- Cholesky decomposition
- iterative methods
- bisection method
- Newton's method
- secant method
- polynomial interpolation
- Aitken's algorithm
- Lagrange interpolation
- Newton interpolation
- Hermite interpolation
- Bezier curves
- De Casteljau's algorithm
- piecewise interpolation
- Spline interpolation
- B-Splines
- Least-squares approximation
- polynomial regression
- difference schemes
- Richardson extrapolation
- Quadrature rules
- Monte Carlo integration
- time stepping schemes for ordinary differential equations
- Runge Kutta schemes
- finite difference method for partial differential equations

Intended Learning Outcomes

By the end of the module, students will be able to

1. describe the basic principles of discretization used in the numerical treatment of continuous problems;
2. 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;
3. recognize mathematical terminology used in textbooks and research papers on numerical methods in the quantitative sciences, engineering, and mathematics to the extent that they fall into the content categories covered in this module;
4. implement simple numerical algorithms in a high-level programming language;
5. understand the documentation of standard numerical library code and understand the potential limitations and caveats of such algorithms.

Indicative Literature

D. Kincaid and W. Cheney (1991). Numerical Analysis: Mathematics of Scientific Computing. Pacific Grove: Brooks/Cole Publishing.

W. Boehm and H. Prautzsch (1993). Numerical Methods. Natick: AK Peters.

Usability and Relationship to other Modules

- This module is a co-recommendation for the module "Applied Dynamical Systems Lab", in which the actual implementation in a high-level programming language of the learned methods will be covered.

Examination Type: Module Examination

Assessment type: Written examination

Duration: 120 min

Weight: 100%

Scope: All intended learning outcomes of this module.

Completion: To pass this module, the examination has to be passed with at least 45%.

8.1.7 Discrete Mathematics

<table>
<tr><td colspan="3"><b>Module Name</b>
Discrete Mathematics</td><td><b>Module Code</b>
CO-501</td><td><b>Level (type)</b>
Year 2 (CORE)</td><td><b>CP</b>
5</td></tr>
<tr><td colspan="6"><b>Module Components</b></td></tr>
<tr><td>Number</td><td colspan="3">Name</td><td>Type</td><td>CP</td></tr>
<tr><td>CO-501-A</td><td colspan="3">Discrete Mathematics</td><td>Lecture</td><td>5</td></tr>
<tr><td><b>Module Coordinator</b>
Dr. Keivan Mallahi-Karai</td><td colspan="3"><b>Program Affiliation</b>
• Mathematics, Modeling and Data Analytics (MMDA)</td><td colspan="2"><b>Mandatory Status</b>
Mandatory for MMDA
Mandatory elective for SDT, CS and RIS</td></tr>
<tr><td colspan="3"><b>Entry Requirements</b></td><td><b>Frequency</b>
Annually (Spring)</td><td colspan="2"><b>Forms of Learning and Teaching</b>
• Lectures (35 hours)
• Private Study (90 hours)</td></tr>
<tr><td>Pre-requisites
☒ None</td><td>Co-requisites
☒ None</td><td>Knowledge, Abilities, or Skills
• Basic university mathematics: can be acquired via the Methods Modules “Calculus and Elements of Linear Algebra I + II” or Matrix Algebra and Advanced Calculus.</td><td><b>Duration</b>
1 semester</td><td colspan="2"><b>Workload</b>
125 hours</td></tr>
<tr><td colspan="6"><b>Recommendations for Preparation</b>
• Some basic familiarity with linear algebra is useful, but not technically required.
• It is recommended to have taken the Methods module: Calculus and Elements of Linear Algebra I + II</td></tr>
<tr><td colspan="6"><b>Content and Educational Aims</b>
This module is an introductory lecture in discrete mathematics. The lecture consists of two main components, enumerative combinatorics and graph theory. The lecture emphasizes connections of discrete mathematics with other areas of mathematics such as linear algebra and basic probability, and outlines applications to areas of computer science, cryptography, etc. where employment of ideas from discrete mathematics has proven to be fruitful. The first part of the lecture—enumerative combinatorics—deals with several classical enumeration problems (Binomial coefficients, Stirling numbers), counting under group actions and generating function. The second half of the lecture—graph theory—includes a discussion of basic notions such as chromatic number, planarity, matchings in graphs, Ramsey theory, and expanders, and their applications.</td></tr>
</table>

Intended Learning Outcomes

By the end of the module, students will be able to

1. demonstrate their mastery of basic tools in discrete mathematics.
2. develop the ability to use discrete mathematics concepts (such as graphs) to model problems in computer science.
3. analyze the definition of basic combinatorial objects such as graphs, permutations, partitions, etc.
4. formulate and design methods and algorithms for solving applied problems based on concepts from discrete mathematics.

Indicative Literature

J.H. van Lint and R.M. Wilson (2001). A Course in Combinatorics, second edition. Cambridge: Cambridge University Press.

B. Bollobas (1998). Modern Graph Theory, Berlin: Springer.

Usability and Relationship to other Modules

- This module is recommended for students pursuing a minor in Mathematics
- This module is a good option as an elective module for students in RIS.

Examination Type: Module Examination

Assessment Type: Written examination

Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of this module

Completion: To pass this module, the examinations must be passed with at least 45%.

8.2 New Skills

8.2.1 Logic (perspective I)

Module Name Logic (perspective I)	**Module Code** CTNS-NSK-01	**Level (type)** Year 2 New Skills	**CP** 2.5

Module Components

Number	Name	Type	CP
CTNS-01	Logic (perspective I)	Lecture (online)	2.5

Module Coordinator Prof. Dr. Jules Coleman	**Program Affiliation** • CONSTRUCTOR Track Area	**Mandatory Status** Mandatory elective for all UG students (one perspective must be chosen)

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites ☒ none	Co-requisites ☒ none	Knowledge, Abilities, or Skills •	Annually (Fall)	Online lecture (17.5h) Private study (45h)
			Duration 1 semester	**Workload** 62.5 hours

Recommendations for Preparation

Content and Educational Aims

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

Students acquire transferable and key skills in this module. By the

end of this module, the students will be able to

1 apply the various principles of logic and expand them to computational thinking.
2 understand the way in which logical processes in humans and in computers are similar and different at the same time.
3 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 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.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written Examination

Duration/Length: 60 min
Weight: 100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.2 Logic (perspective II)

Module Name			**Module Code**	**Level (type)**	**CP**
Logic (perspective II)			CTNS-NSK-02	Year 2 New Skills	2.5
Module Components					
Number	Name			Type	CP
CTNS-02	Logic (perspective II)			Lecture (online)	2.5
Module Coordinator	**Program Affiliation**			**Mandatory Status**	
NN	• CONSTRUCTOR Track Area			Mandatory elective for all UG students (one perspective must be chosen)	
Entry Requirements			**Frequency**	**Forms of Learning and Teaching**	
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall)	Online lecture (17.5h) Private study (45h)	
☒ none	☒ none	•			
			Duration	**Workload**	
			1 semester	62.5 hours	
Recommendations for Preparation					

Content and Educational Aims

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. Once 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

Students acquire transferable and key skills in this module. By the

end of this module, the students will be able to

1. apply the various principles of logic
2. explain practical relevance of non-standard logic
3. describe how many-valued logic extends basic predicate logic
4. apply basic rules of fuzzy logic to calculate partial truth values
5. sketch basic rules of temporal logic
6. implement predicates in a logic programming language
7. 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.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written Examination

Duration/Length: 60 min

Weight: 100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination must be passed with at least 45%.

8.2.3 Causation and Correlation (perspective I)

Module Name	**Module Code**	**Level (type)**	**CP**
Causation and Correlation (perspective I)	CTNS-NSK-03	Year 2 New Skills	2.5

Module Components

Number	Name	Type	CP
CTNS-03	Causation and Correlation	Lecture (online)	2.5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
Prof. Dr. Jules Coleman	• CONSTRUCTOR Track Area	Mandatory elective for all UG students (one perspective must be chosen)

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	**Frequency**	**Forms of Learning and Teaching**
☒ none	☒ none		Annually (Spring)	Online lecture (17.5h) Private study (45h)
			Duration	**Workload**
			1 semester	62.5 hours

Recommendations for Preparation

Content and Educational Aims

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

Students acquire transferable and key skills in this module. By the

end of this module, the students will be able to

1. formulate testable hypotheses that are designed to reveal causal connections and those designed to reveal interesting, important and useful correlations.
2. distinguish scientifically interesting correlations from unimportant ones.
3. apply critical thinking skills to evaluate information.
4. 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.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written Examination

Duration/Length: 60 min
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.4 Causation and Correlation (perspective II)

Module Name Causation and Correlation (perspective II)	**Module Code** CTNS-NSK-04	**Level (type)** Year 2 New Skills	**CP** 2.5

Module Components

Number	Name	Type	CP
CTNS-04	Causation and Correlations	Lecture (online)	2.5

Module Coordinator Dr. Keivan Mallahi-Karai, Dr. Eoin Ryan, Dr. Irina Chiaburu	**Program Affiliation** • CONSTRUCTOR Track Area	**Mandatory Status** Mandatory elective for all UG students (one perspective must be chosen)

Entry Requirements Pre-requisites ☒ none	Co-requisites ☒ none	Knowledge, Abilities, or Skills • Basic probability theory	**Frequency** Annually (Spring)	**Forms of Learning and Teaching** Online lecture (17.5h) Private study (45h
			Duration 1 semester	**Workload** 62.5 hours

Recommendations for Preparation

Content and Educational Aims

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.

Intended Learning Outcomes

Students acquire transferable and key skills in this module.

1. By the end of this module, the students will be able to
2. have a clear understanding of the history of causal thinking. form a critical understanding of the key debates and controversies surrounding the idea of causality.
3. recognize and apply probabilistic causal models.
4. explain how understanding of causality differs among different disciplines.
5. 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.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment: Written examination

Duration/Length: 60 min

Weight: 100 %

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.5 Linear Model and Matrices

Module Name Linear Model and Matrices	**Module Code** CTNS-NSK-05	**Level (type)** Year 3 New Skills	**CP** 5

Module Components

Number	Name	Type	CP
CTNS-05	Linear models and matrix	Seminar	5

Module Coordinator Prof. Dr. Marc-Thorsten Hütt	**Program Affiliation** • CONSTRUCTOR Track Area	**Mandatory Status** Mandatory elective

Entry Requirements

Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills
Logic Causation & Correlation	☒ none	

Frequency Annually (Fall)	**Forms of Learning and Teaching** Online lecture (35h) Private Study (90h)	
Duration 1 Semester	**Workload** 125 hours	

Recommendations for Preparation

Content and Educational Aims

There are no universal 'right skills'. But the notion of linear models and the avenue to matrices and their properties can be useful in diverse disciplines to implement a quantitative, computational approach. Some of the most popular data and systems analysis strategies are built upon this framework. Examples include principal component analysis (PCA), the optimization techniques used in Operations Research (OR), the assessment of stable and unstable states in nonlinear dynamical systems, as well as aspects of machine learning.

Here we introduce the toolbox of linear models and matrix-based methods embedded in a wide range of transdisciplinary applications (part 1). We describe its foundation in linear algebra (part 2) and the range of tools and methods derived from this conceptual framework (part 3). At the end of the course, we outline applications to graph theory and machine learning (part 4). Matrices can be useful representations of networks and of system of linear equations. They are also the core object of linear stability analysis, an approach used in nonlinear dynamics. Throughout the course, examples from neuroscience, social sciences, medicine, biology, physics, chemistry, and other fields are used to illustrate these methods.

A strong emphasis of the course is on the sensible usage of linear approaches in a nonlinear world. We will critically reflect the advantages as well as the disadvantages and limitations of this method. Guiding questions are: How appropriate is a linear approximation of a nonlinear system? What do you really learn from PCA? How reliable are the optimal states obtained via linear programming (LP) techniques?

This debate is embedded in a broader context: How does the choice of a mathematical technique confine your view on the system at hand? How, on the other hand, does it increase your capabilities of analyzing the system (due to software available for this technique, the ability to compare with findings from other fields built upon the same technique and the volume of knowledge about this technique)?

In the end, students will have a clearer understanding of linear models and matrix approaches in their own discipline, but they will also see the full transdisciplinarity of this topic. They will make better decisions in their choice of data analysis methods and become mindful of the challenges when going from linear to nonlinear thinking.

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. apply the concept of linear modeling in their own discipline
2. distinguish between linear and nonlinear interpretation strategies and understand the range of applicability of linear models
3. make use of data analysis / data interpretation strategies from other disciplines, which are derived from linear algebra
4. be aware of the ties that linear models have to machine learning and network theory

Note that these four ILOs can be loosely associated with the four parts of the course indicated above

Indicative Literature

Part 1:

material from Linear Algebra for Everyone, Gilbert Strang, Wellesley-Cambridge Press, 2020 Part 2:

material from Introduction to Linear Algebra (5th Edition), Gilbert Strang, Cambridge University Press, 2021 Part 3:

Mainzer, Klaus. "Introduction: from linear to nonlinear thinking." Thinking in Complexity: The Computational Dynamics of Matter, Mind and Mankind (2007): 1-16.

material from Mathematics of Big Data: Spreadsheets, Databases, Matrices, and Graphs, Jeremy Kepner, Hayden Jananthan, The MIT Press, 2018

material from Introduction to Linear Algebra (5th Edition), Gilbert Strang, Cambridge University Press, 2021 Part 4:

material from Linear Algebra and Learning from Data, Gilbert Strang, Wellesley-Cambridge Press, 2019

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment: Written examination

Duration/Length: 120 min

Weight: 100 %

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.6 Complex Problem Solving

Module Name		**Module Code**	**Level (type)**	**CP**
Complex Problem Solving		CTNS-NSK-06	Year 3 New Skills	5
Module Components				
Number	Name		Type	CP
CTNS-06	Complex Problem Solving		Lecture (online)	5
Module Coordinator	**Program Affiliation**		**Mandatory Status**	
Prof. Dr. Marco Verweij	• CONSTRUCTOR Track Area		Mandatory elective	

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall)	Online Lectures (35h) Private Study (90h)
Logic Causation & Correlation	☒ none	• Project Management • Complex Problem Solving	**Duration**	**Workload**
			1 semester	125 hours

Recommendations for Preparation

Wherever possible intuition will be emphasized over technical detail. Technical readings will be made available and discussed with students in class.

Content and Educational Aims

Complex problems are, by definition, non-linear and/or emergent. Some fifty years ago, scholars such as Herbert Simon began to argue that societies around the world had developed an impressive array of tools with which to solve simple and even complicated problems, but still needed to develop methods with which to address the rapidly increasing number of complex issues. Since then, a variety of such methods has emerged. These include 'serious games' developed in computer science, 'multisector systems analysis' applied in civil and environmental engineering, 'robust decision-making' proposed by the RAND Corporation, 'design thinking' developed in engineering and business studies, 'structured problem solving' used by McKinsey & Co., 'real-time technology assessment' advocated in science and technology studies, and 'deliberative decision-making' emanating from political science.

In this course, students first learn to distinguish between simple, complicated and complex problems. They also become familiar with the ways in which a particular issue can sometimes shift from one category into another. In addition, the participants learn to apply several tools for resolving complex problems. Finally, the students are introduced to the various ways in which natural and social scientists can help stakeholders resolve complex problems. Throughout the course examples and applications will be used. When possible, guest lectures will be offered by experts on a particular tool for tackling complex issues. For the written, take-home exam, students will have to select a specific complex problem, analyse it and come up with a recommendation – in addition to answering several questions about the material learned.

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. Identify a complex problem;
2. Develop an acceptable recommendation for resolving complex problems.
3. Understand the roles that natural and social scientists can play in helping stakeholders resolve complex problems;

Indicative Literature

Camillus, J. (2008). Strategy as a wicked problem. Harvard Business Review 86: 99-106.

Chia, A. (2019). Distilling the essence of the McKinsey way: The problem-solving cycle. Management Teaching Review 4(4): 350-377.

Den Haan, J., van der Voort, M.C., Baart, F., Berends, K.D., van den Berg, M.C., Straatsma, M.W., Geenen, A.J.P., & Hulscher, S.J.M.H. (2020). The virtual river game: Gaming using models to collaboratively explore river management complexity, Environmental Modelling & Software 134, 104855,

Folke, C., Carpenter, S., Elmqvist, T., Gunderson, L., Holling, C.S., & Walker, B. (2002). Resilience and sustainable development: Building adaptive capacity in a world of transformations. AMBIO: A Journal of the Human Environment 31(5): 437-440.

Ostrom, E. (2010). Beyond markets and states: Polycentric governance of complex economic systems. American Economic Review 100(3): 641-72.

Pielke, R. Jr. (2007). The honest broker: Making sense of science in policy and politics. Cambridge: Cambridge University Press.

Project Management Institute (2021). A guide to the project management body of knowledge (PMBOK® guide). Schon, D.

A., & Rein, M. (1994). Frame reflection: Toward the resolution of intractable policy controversies. New York: Basic Books.

Simon, H. A. (1973). The structure of ill structured problems. Artificial Intelligence 4(3-4): 181-201.

Verweij, M. & Thompson, M. (Eds.) (2006). Clumsy solutions for a complex world. London: Palgrave Macmillan.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written examination Duration: 120 min Weight:

100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.7 Argumentation, Data Visualization and Communication (perspective I)

Module Name Argumentation, Data Visualization and Communication (perspective I)	**Module Code** CTNS-NSK-07	**Level (type)** Year 3 New Skills	**CP** 5

Module Components

Number	Name	Type	CP
CTNS-07	Argumentation, Data Visualization and Communication	Lecture (online)	5

Module Coordinator Prof. Dr. Jules Coleman, Prof. Dr. Arvid Kappas	**Program Affiliation** • CONSTRUCTOR Track Area	**Mandatory Status** Mandatory elective for all UG students (one perspective must be chosen)

Entry Requirements Pre-requisites Logic Causation & Correlation	Co-requisites ☒ none	Knowledge, Abilities, or Skills	**Frequency** Annually (Fall)	**Forms of Learning and Teaching** Online Lectures (35h) Private Study (90h)
			Duration 1 semester	**Workload** 125h

Recommendations for Preparation

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

Students acquire transferable and key skills in this module. By the end of this module, the students will be able to

1. Distinguish among different modalities of argument, e.g. legal arguments, vs. scientific ones.
2. Construct arguments using tools of data visualization.
3. 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.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written Examination

Duration/Length: 120 (min)
Weight: 100%

Scope: All intended learning outcomes of the module

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.8 Argumentation, Data Visualization and Communication (perspective II)

Module Name Argumentation, Data Visualization and Communication (perspective II)			**Module Code** CTNS-NSK-08	**Level (type)** Year 3 New Skills	**CP** 5
Module Components					
Number	Name			Type	CP
CTNS-08	Communication, Interaction, and Argumentation			Lecture (online)	5
Module Coordinator Prof. Dr. Jules Coleman, Prof. Dr. Arvid Kappas	**Program Affiliation** • CONSTRUCTOR Track Area			**Mandatory Status** Mandatory elective for all UG students (one perspective must be chosen)	
Entry Requirements			**Frequency**	**Forms of Learning and Teaching**	
Pre-requisites Logic Causation & Correlation	Co-requisites ☒ none	Knowledge, Abilities, or Skills • ability and openness to engage in interactions • media literacy, critical thinking and a proficient handling of data sources • own research in academic literature	annually	• Lecture (35 hours) • Tutorial of the lecture (10 hours) • Private study for the lecture (80 hours)	
			Duration 1 semester	**Workload** 125 hours	
Recommendations for Preparation					

Content and Educational Aims

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.

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. Analyze communication processes in formal and informal contexts.
2. Identify challenges and failures in communication.
3. Design communications to achieve specified goals to specific target groups.
4. Understand the principles of argumentation theory.
5. 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

Examination Type: Module Examination

Assessment Type: Digital submission of asynchronous presentation, including reflection Duration/Length:

Asynchronous/Digital submission

Weight: 100%

Scope: All intended learning outcomes of the module

Module achievement: 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

8.2.9 Agency, Leadership, and Accountability

Module Name Agency, Leadership, and Accountability		**Module Code** CTNS-NSK-09	**Level (type)** Year 3 New Skills	**CP** 5
Module Components				
Number	Name		Type	CP
CTNS-09	Agency, Leadership, and Accountability		Lecture (online)	5
Module Coordinator Prof. Dr. Jules Coleman	**Program Affiliation** • CONSTRUCTOR Track Area		**Mandatory Status** Mandatory for CSSE Mandatory elective for all other UG study programs	
Entry Requirements Pre-requisites ☒ none	Co-requisites ☒ none Knowledge, Abilities, or Skills	**Frequency** Annually (Spring)	**Forms of Learning and Teaching** Online Lectures (35h) Private Study (90h)	
		Duration	**Workload** 125 hours	
Recommendations for Preparation				

Content and Educational Aims

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

Students acquire transferable and key skills in this module. By the

end of this module, the students will be able to

1. 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 how science is an economic sector, populated by large powerful organizations that set norms, fund research agendas.
3. 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 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).

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written examination	Duration/Length: 120 min Weight: 100%
Scope: All intended learning outcomes of the module	

Completion: To pass this module, the examination has to be passed with at least 45%

8.2.10 Community Impact Project

Module Name	**Module Code**	**Level (type)**	**CP**
Community Impact Project	CTNC-CIP-10	Year 3 New Skills	5

Module Components			
Number	Name	Type	CP
CTNC-10	Community Impact Project	Project	5

Module Coordinator	**Program Affiliation**	**Mandatory Status**
CIP Faculty Coordinator	• CONSTRUCTOR Track Area	Mandatory elective

Entry Requirements			**Frequency**	**Forms of Learning and Teaching**
Pre-requisites	Co-requisites	Knowledge, Abilities, or Skills	Annually (Fall / Spring)	• Introductory, accompanying, and final events: 10 hours • Self-organized teamwork and/or practical work in the community: 115 hours
☒ at least 15 CP from CORE modules in the major	☒ None	• Basic knowledge of the main concepts and methodological instruments of the respective disciplines	**Duration**	**Workload**
			1 semester	125 hours

Recommendations for Preparation

Develop or join a community impact project before the 5th or 6th semester based on the introductory events during the 4th semester by using the database of projects, communicating with fellow students and faculty, and finding potential companies, organizations, or communities to target.

Content and Educational Aims

CIPs are self-organized, major-related, and problem-centered applications of students' acquired knowledge and skills. These activities will ideally be connected to their majors so that they will challenge the students' sense of practical relevance and social responsibility within the field of their studies. Projects will tackle real issues in their direct and/or broader social environment. These projects ideally connect the campus community to other communities, companies, or organizations in a mutually beneficial way.

Students are encouraged to create their own projects and find partners (e.g., companies, schools, NGOs), but will get help from the CIP faculty coordinator team and faculty mentors to do so. They can join and collaborate in interdisciplinary groups that attack a given issue from different disciplinary perspectives.

Student activities are self-organized but can draw on the support and guidance of both faculty and the CIP faculty coordinator team.

Intended Learning Outcomes

The Community Impact Project is designed to convey the required personal and social competencies for enabling students to finish their studies at Jacobs as socially conscious and responsible graduates (part of the Constructor University'smission) and to convey social and personal abilities to the students, including a practical awareness of the societal context and relevance of their academic discipline.

By the end of this project, students will be able to

- understand the real-life issues of communities, organizations, and industries and relate them to concepts in their own discipline;
- enhance problem-solving skills and develop critical faculty, create solutions to problems, and communicate these solutions appropriately to their audience;
- apply media and communication skills in diverse and non-peer social contexts;
- develop an awareness of the societal relevance of their own scientific actions and a sense of social responsibility for their social surroundings;
- reflect on their own behavior critically in relation to social expectations and consequences;
- work in a team and deal with diversity, develop cooperation and conflict skills, and strengthen their empathy and tolerance for ambiguity.

Indicative Literature

Usability and Relationship to other Modules

- Students who have accomplished their CIP (6th semester) are encouraged to support their fellow students during the development phase of the next year’s projects (4th semester).

Examination Type: Module Examination

Project, not numerically graded (pass/fail)
Scope: All intended learning outcomes of the module

8.3 Language and Humanities Modules

The descriptions of the language modules are provided in a separate document, the “Language Module Handbook” that can be accessed from the Constructor University’s Language & Community Center internet sites https://constructor.university/student-life/language- community-center.

8.3.1 Humanities

8.3.1.1 Introduction to Philosophical Ethics

Module Name Introduction to Philosophical Ethics			**Module Code** CTHU-HUM- 001	**Level (type)** Year 1	**CP** 2.5
Module Components					
Number	Name			Type	CP
CTHU-001	Introduction to Philosophical Ethics			Lecture (online)	2.5
Module Coordinator Dr. Eoin Ryan	**Program Affiliation** • CONSTRUCTOR Track Area			**Mandatory Status** Mandatory elective	
Entry Requirements			**Frequency** Annually (Fall or Spring)	**Forms of Learning and Teaching** Online lectures (17.5 h) Private Study (45h)	
Pre-requisites ☒ none	Co-requisites ☒ none	Knowledge, Abilities, or Skills •	**Duration** 1 semester	**Workload** 62.5 hours	
Recommendations for Preparation					
Content and Educational Aims The nature of morality – how to lead a life that is good for yourself, and how to be good towards others – has been a central debate in philosophy since the time of Socrates, and it is a topic that continues to be vigorously discussed. This course will introduce students to some of the key aspects of philosophical ethics, including leading normative theories of ethics (e.g. consequentialism or utilitarianism, deontology, virtue ethics, natural law ethics, egoism) as well as some important questions from metaethics (are useful and generalizable ethical claims even possible; what do ethical speech and ethical judgements actually do or explain) and moral psychology (how do abstract ethical principles do when realized by human psychologies). The course will describe ideas that are key factors in ethics (free will, happiness, responsibility, good, evil, religion, rights) and indicate various routes to progress in understanding ethics, as well as some of their difficulties.					

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. Describe normative ethical theories such as consequentialism, deontology and virtue ethics.
2. Discuss some metaethical concerns.
3. Analyze ethical language.
4. Highlight complexities and contradictions in typical ethical commitments.
5. Indicate common parameters for ethical discussions at individual and social levels.
6. Analyze notions such as objectivity, subjectivity, universality, pluralism, value.

Indicative Literature

Simon Blackburn, Being Good (2009)

Russ Shafer-Landay, A Concise Introduction to Ethics (2019)

Mark van Roojen, Metaethicas: A Contemporary Introduction (2015)

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment Type: Written Examination

Duration/Length: 60 min
Weight: 100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%.

8.3.1.2 Introduction to the Philosophy of Science

Module Name Introduction to the Philosophy of Science			**Module Code** CTHU-HUM- 002	**Level (type)** Year 1	**CP** 2.5
Module Components					
Number	Name			Type	CP
CTHU-002	Introduction to the Philosophy of Science			Lecture (online)	2.5
Module Coordinator Dr. Eoin Ryan	**Program Affiliation** • CONSTRUCTOR Track Area			**Mandatory Status** Mandatory elective	
Entry Requirements Pre-requisites ☒ none	Co-requisites ☒ none	Knowledge, Abilities, or Skills	**Frequency** Annually (Fall or Spring)	**Forms of Learning and Teaching** Online lectures (17.5h) Private Study (45h)	
			Duration 1 semester	**Workload** 62.5 hours	
Recommendations for Preparation					
Content and Educational Aims This humanities module will introduce students to some of the central ideas in philosophy of science. Topics will include distinguishing science from pseudo-science, types of inference and the problem of induction, the pros and cons of realism and anti-realism, the role of explanation, the nature of scientific change, the difference between natural and social sciences, scientism and the values of science, as well as some examples from philosophy of the special sciences (e.g., physics, biology). The course aims to give students an understanding of how science produces knowledge, and some of the various contexts and issues which mean this process is never entirely transparent, neutral, or unproblematic. Students will gain a critical understanding of science as a human practice and technology; this will enable them both to better understand the importance and success of science, but also how to properly critique science when appropriate.					
Intended Learning Outcomes Upon completion of this module, students will be able to 1. Understand key ideas from the philosophy of science. 2. Discuss different types of inference and rational processes. 3. Describe differences between how the natural sciences, social sciences and humanities discover knowledge. 4. Identify ways in which science can be more and less value-laden. 5. Illustrate some important conceptual leaps in the history of science.					
Indicative Literature Peter Godfrey-Smith, Theory and Reality (2021) James Ladyman, Understanding Philosophy of Science (2002) Paul Song, Philosophy of Science: Perspectives from Scientists (2022)					
Usability and Relationship to other Modules					

Examination Type: Module Examination

Assessment Type: Written Examination	Duration/Length: 60 min Weight: 100%

Scope: All intended learning outcomes of the module.

Completion: To pass this module, the examination has to be passed with at least 45%.

8.3.1.3 Introduction to Visual Culture

Module Name Introduction to Visual Culture			**Module Code** CTHU-HUM- 003	**Level (type)** Year 1	**CP** 2.5
Module Components					
Number	Name			Type	CP
CTHU-003	Introduction to Visual Culture			Lecture (online)	2.5
Module Coordinator Dr. Irina Chiaburu	**Program Affiliation** • CONSTRUCTOR Track Area			**Mandatory Status** Mandatory elective	
Entry Requirements Pre-requisites ☒ none	Co-requisites ☒ none	Knowledge, Abilities, or Skills •	**Frequency** Annually (Spring/Fall)	**Forms of Learning and Teaching** Online Lecture	
			Duration 1 semester	**Workload** 62.5 h	
Recommendations for Preparation					

Content and Educational Aims

Of the five senses, the sense of sight has for a long time occupied the central position in human cultures. As John Berger has suggested this could be because we can see and recognize the world around us before we learn how to speak. Images have been with us since the earliest days of the human history. In fact, the earliest records of human history are images found on cave walls across the world. We use images to capture abstract ideas, to catalogue and organize the world, to represent the world, to capture specific moments, to trace time and change, to tell stories, to express feelings, to better understand, to provide evidence and more. At the same time, images exert their power on us, seducing us into believing in their ‘innocence’, that is into forgetting that as representations they are also interpretations, i.e., a particular version of the world.

The purpose of this course is to explore multiple ways in which images and the visual in general mediate and structure human experiences and practices from more specialized discourses, e.g., scientific discourses, to more informal and personal day-to-day practices, such as self-fashioning in cyberspace. We will look at how social and historical contexts affect how we see, as well as what is visible and what is not. We will explore the centrality of the visual to the intellectual activity, from early genres of scientific drawing to visualizations of big data. We will examine whether one can speak of visual culture of protest, look at the relationship between looking and subjectivity and, most importantly, ponder the relationship between the visual and the real.

Intended Learning Outcomes

Upon completion of this module, students will be able to

1. Understand a range of key concepts pertaining to visual culture, art theory and cultural analysis
2. Understand the role visuality plays in development and maintenance of political, social, and intellectual discourses
3. Think critically about images and their contexts
4. Reflect critically on the connection between seeing and knowing

Indicative Literature

Berger, J., Blomberg, S., Fox, C., Dibb, M., & Hollis, R. (1973). Ways of seeing.

Foucault, M. (2002). The order of things: an archaeology of the human sciences (Ser. Routledge classics). Routledge.

Hunt, L. (2004). Politics, culture, and class in the French revolution: twentieth anniversary edition, with a new preface (Ser. Studies on the history of society and culture, 1). University of California Press.

Miller, V. (2020). Understanding digital culture (Second). SAGE.

Thomas, N. (1994). Colonialism's culture: anthropology, travel and government. Polity Press.

Usability and Relationship to other Modules

Examination Type: Module Examination

Assessment: Written examination Duration/Length: 60 min.

Weight: 100%

Scope: all intended learning outcomes

Completion: To pass this module, the examination has to be passed with at least 45%.

9 Appendix

9.1 Intended Learning Outcomes Assessment-Matrix

Robotics and Intelligent Systems					Introduction to Robotics and Intelligent Systems	Algorithms and Data Structures	Programming in C and C++	Classical Physics	General Electrical Engineering I	Inbtroduction to Computer Science	Robotics	Machine Learning	RIS Lab	Automation	Embedded Systems	Control Systems	Computer Vision	Artificial Intelligence	RIS project	Marine Robotics	Human-Robot Interaction	Optimization	Calculus and Linear Algebra 1	Calculus and Linear Algebra 2	Probability and Random Processes	Discrete Mathematics / Numerical Methods	Internship	CT New Skills	CT German Language and Humanities	Bachelor Thesis
Semester					2	2	1	1	1	2	3	4	3/4	4	3	3	3	4	4	5/6	5	5/6	1	2	3	4	5	3-6	1-2	6
mandatory/mandatory elective					m	m	m	m	m	m	m	m	me	me	me	me	me	m	m	me	me	me	m	m	m	me	m	me	me	m
Credits					7.5	7.5	7.5	7.5	7.5	7.5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	15	20	5	15
	Competencies*																													
Program Learning Outcomes	A	E	P	S																										
design basic electronics circuits	x	x				x	x			x																				
think in an analytic way at multiple levels of abstraction	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		x		
develop, analyze and implement algorithms using modern software engineering methods.	x	x				x	x			x							x		x	x	x									
demonstrate knowledge of kinematics and dynamics of multi-body systems	x	x			x						x																			
design and develop linear and nonlinear control systems	x	x														x			x											
design basic electronics circuits	x	x			x				x						x															
examine physical problems, apply mathematical skills to find possible solutions and assess them critically	x	x						x						x																
show competence about operational principles of motors and drives	x	x												x					x											
design and develop machine learning algorithms and techniques for pattern-recognition, classification, and decision-making under uncertainty;	x	x										x							x		x									
design and develop computer vision algorithms for inferring 3D information from camera images, and for object recognition and localization	x	x															x		x		x									
model common mechanical and electrical systems that are part of intelligent mobile systems	x	x			x				x				x																	
design robotics systems and program them using popular robotics software frameworks	x	x									x		x						x	x	x									
use academic or scientific methods as appropriate in the field of Robotics and Intelligent Systems 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	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	x	x					x			x
engage ethically with the 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	x	x	x	x	x	x	x	x	x	x	x	x	x	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	x	x	x
apply their knowledge and understanding to a professional context;	x	x	x								x	x	x	x	x	x	x	x	x	x		x								
work effectively in a diverse team and take responsibility in a team;	x		x		x									x					x	x	x						x			
adhere to and defend ethical, scientific, and professional standards.	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	x	x	x
Assessment Type																														
Oral examination																				x										
Written examination					x	x	x	x	x	x	x	x		x		x	x	x				x	x	x	x	x		x		
Project															x				x									x		x
Term paper																											x			
Report								x	x				x														x			
Poster presentation																														x
Presentation																			x									x		
Various																													x	
Practical assessment										x																				
module achievements/bonus achievements					x	x	x										x											x	x	
																					x									

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

Figure 3: Intended Learning Outcomes Assessment-Matrix