Course Descriptions International Computer Science

Summer Semester 2025

13 February 2025

German (different course levels)

Course title	see schedule Language Centre
ECTS	4
Course type	Seminar
SWS	4
Semester	Winter and summer
Workload in hours	60 hrs
Assessment method	Written examination, 90 min.
Language of instruction	German

Please find here the course descriptions for German language courses at all course levels:
https://th-deg.de/en/students/language-electives#german

English in Technical Contexts B2

Course title	English in Technical Contexts B2
ECTS	2
Course type	Language training course
SWS	2
Semester	Winter and summer
Course level	**B2** • Can understand the main ideas of complex text on both concrete and abstract topics, including technical discussions in his/her field of specialization • Can interact with a degree of fluency and spontaneity that makes regular interaction with native speakers quite possible without strain for either party • Can produce clear, detailed text on a wide range of subjects and explain a viewpoint on a topical issue giving the advantages and disadvantages of various options
Lecturer	Neal O'Donoghue, MA
Course objectives	This course aims to deepen students' encounter with the English language in a technical context by giving practical training in specialized vocabulary, grammar and language usage. The four cardinal language skills – listening, speaking, reading, and writing – will play an integral role in this training. The course is designed to be relevant and interesting for engineering students and will be adapted to their learning needs and study areas.

	By the end of the course, participants should have a more comprehensive understanding of, and enhanced fluency in, the English language in an engineering context.
Course contents	*Obligatory topics (60 %):* • Numbers and mathematical operations • Shapes and dimensions • August 2017 • Basic physics and the scientific worldview • Materials and their properties • Case study on an area related to technology • /physics/engineering • Grammar/ communication skills *Variable content (40 %):* Variable content will be determined on the basis of a student survey conducted in the first session. Current world events (including news events and popular culture) and recent technological innovations may be used as a basis for discussions.
Teaching methods	Teaching methods focus on improving the four cardinal language skills and include group discussions and group projects; individual work; mini-presentations; role-plays; close reading and listening activities; dictation; grammar games; and various follow-up viewing and writing activities. Work not completed in class should be done at home. Self-study assignments will be set on a weekly basis.
Assessment method	Written exam (60 min) No dictionaries are allowed. Exam structure: • Part 1: Listening comprehension(s) • Part 2: Reading comprehension(s)

	• Part 3: Vocabulary and technical content • Part 4: Grammar (maximum 10% of total exam points, excluding writing exercise) • Part 5: Writing composition (150-200 words) The exam will be based on topics covered during the semester. The exam will be based on topics covered during the semester.
Recommended Literature	Astley, Peter, and Lewis Lansford. Engineering 1: Student's Book. Oxford: Oxford UP, 2013. Print. Bauer, Hans-Jürgen. English for Technical Purposes. Berlin: Cornelsen, 2000. Print. Bonamy, David. Technical English 4. Harlow, England: Pearson Education, 2011. Print. Bonamy, David, and Christopher Jacques. Technical English 3. Harlow: Pearson Longman, 2011. Print. Brieger, Nick, and Alison Pohl. Technical English: Vocabulary and Grammar. Oxford: Summertown, 2002. Print. Dummett, Paul. Energy English: For the Gas and Electricity Industries. Hampshire: Heinle, Cengage Learning, 2010. Print. Dunn, Marian, David Howey, and Amanda Ilic. English for Mechanical Engineering in Higher Education Studies Coursebook. Reading: Garnet Education, 2010. Print. engine: Englisch für Ingenieure. <www.engine-magazin.de> (Darmstadt). Various issues. Print. Foley, Mark, and Diane Hall. MyGrammarLab. Harlow: Pearson, 2012. Print. Glendinning, Eric H., and Norman Glendinning. Oxford English for Electrical and Mechanical Engineering. Oxford: Oxford UP, 1995. Print.

	Glendinning, Eric H., and Alison Pohl. Technology 2. Oxford: Oxford UP, 2008. Print. Heidenreich, Sharon. English for Architects and Civil Engineers. Wiesbaden: Vieweg + Teubner Verlag, 2008. Print. Ibbotson, Mark. Cambridge English for Engineering. Cambridge: Cambridge UP, 2008. Print. Ibbotson, Mark. Professional English in Use. Engineering: Technical English for Professionals. Cambridge: Cambridge UP, 2009. Print. Markner-Jäger, Brigitte. Technical English: Civil Engineering and Construction. Haan-Gruiten: Verl. Europa-Lehrmittel, 2013. Print. Murphy, Raymond. English Grammar in Use. Cambridge: Cambridge UP, 2004. Print. Schäfer, Wolfgang. Construction Milestones: Englisch Für Bau-, Holz- Und Anlagenberufe. Stuttgart: Klett, 2013. Print. Wagner, Georg, and Maureen Lloyd. Zörner. Technical Grammar and Vocabulary: A Practice Book for Foreign Students. Berlin: Cornelsen, 1998. Print.
Language of instruction	English
Prerequisites	B1 / Abitur (A-levels/ school leaving certificate giving right of entry to higher education) / 7-9 years of English

Intercultural Training for Germany and Bavaria

Course title	Intercultural Training for Germany and Bavaria
ECTS	1
Course type	Elective
SWS	1
Semester	Winter and summer
Workload in hours	30 hours
Name of Instructor	Lisa Werner
Course objectives	Participants get an understanding of the different theories of "culture" and learn about stereotypes and traditions in Bavaria. Furthermore, the participants get information on Germany and Bavaria as well as the Deggendorf Institute of Technology.
Course contents	I. Culture (theroies) II. Customs and Rituals in Germany/Bavaria III. Information on Germany and Bavaria and the DIT IV. Quiz and Presentation V. Culture Shock
Recommended literature	Bolten J. und Ehrhardt C., Interkulturelle Kommunikation, Verlag Wissenschaft & Praxis 2003; Bolten J, Einführung in die interkulturelle Wirtschaftskommunikation, Vandenhoeck & Ruprecht 2007
Teaching methods	The course is organized according to four pillars:

	1. Culture 2. Customs and Rituals 3. Information on Germany/Bavaria 4. Culture Shock Whereas hard facts are taught in a classical lecture style, students will do lots of role-plays, critical incidents, short movies and do a quiz.
Assessment method	Paper
Language of instruction	English/German
Prerequisites	None

Bavarian Culture

Course title	Bavarian Culture
Course ID	229
SWS	2
Semester	Winter and summer
ECTS	2
Course type	Elective
Language of instruction	English
Name of lecturer	Jennifer Hauer
Course objectives	Participants get a deeper understanding of the traditional and contemporary Bavarian culture by integrating knowledge about customs, language, and history with culturally routed events.
Course contents	1. Hard facts 1.1. History 1.2. Demographics 1.3. Geography 2. Customs and rituals 2.1. Traditional 2.2. Contemporary 3. Language 4. Events
Teaching methods	The course is organized according to four pillars: 1. Hard Facts 2. Customs and Rituals 3. Language 4. Events

	Whereas hard facts are taught in a classical lecture style, students should experience aspects of the culture in a lively manner through knowledge dissemination of cultural experts, off-campus seminars at events of traditional cultural origin, as well as learning and engaging in cultural rituals themselves. The aim is to deepen and complement the contents taught in the Orientation Week.
Recommended literature	Jonas, B., Gebrauchsanweisung für Bayern, Piper Verlag, 2007
Assessment methods	Seminar paper
Prerequisites	Participants should have attended the introductory Intercultural Training during the Orientation Week.

Business and Society in China & Emerging Asia

Course title	Business and Society in China & Emerging Asia
ECTS	2
Course type	Elective
SWS	2
Semester	Summer
Workload in hours	Total: 60 / In-class: 30 / Self-study: 30
Lecturer	Prof. Dr. Wei Manske-Wang
Course objectives	• Awareness of foreign cultures and understanding their causes • Think out of the box and establish global horizons • Preparing for the challenges of future professional life in a global environment • Doing business in China/Asia successfully requires a holistic view on China/Asia and a thorough understanding how business is done there! This course aims at providing students with the necessary knowledge about contextual determinants of business practice (culture, politics, economy, society, history) and introduces exemplary reference cases.
Course contents	• The historical roots of China: What are structural legacies of the past? How do Chinese perceptions of history influence the present society? • The institutional setting of the Chinese economy: What are the main actors in the Chinese economy (state-owned enterprises, private-owned businesses)?

- The political system and its ramifications in the domain of economic policy and business: What is the role of the Communist Party? What are the principal decision makers on different levels of government? How does this affect central aspects of business environment such as corporate governance?
- What is behind Chinese long-term strategy “Belt and road initiative”?
- Culture and societal values: China represents an amazing mix of global metropolitan life and a resurgence of tradition, deeply enmeshed in her high-speed urbanization process that continue shaping the country in the last decades.
- What do you know about Chinese philosophies in the past? What do you know about Chinese values today?
- What are implications for business, such as regarding consumer demand of young generation?
- Behavioural aspects of business practice: The Chinese are famous for networking. We look at the ‘Chinese way’ in establishing social relations in the business domain. Further, we explore Chinese organizational behaviour in companies.
- What are ‘mega-trends’ of the future affecting the outlook for Chinese business? We touch on issues such as demographic change, looming environmental crises, digitalization and the question of political stability.
- Institutions and strategic arrangements in Asia: ASEAN, APEC, BRICS, BRI, RCEP etc.
- More countries in Asia: Japan, India, Vietnam, Indonesia etc.
- Is an Asian Century dawning?

Recommended literature	Hofstede, G.; Hofstede G.J. (2009): Lokales Denken, globales Handeln: Interkulturelle Zusammenarbeit und globales Management. 4. Auflage. München: Deutscher Taschenbuch Verlag Thomas, A.; Kammhuber S.; Schroll-Machl, S. (Hg.) (2007): Handbuch Interkulturelle Kommunikation und Kooperation Band 2: Länder, Kulturen und interkulturelle Berufstätigkeit. 2. Auflage. Göttingen: Vandenhoeck & Ruprecht
Teaching methods	Lecture, Press Monitoring, Case Studies, Discussions, Group Work, Q&A
Assessment method	Group works – Written Assignment (50%) & Final Presentation (50%)
Language of instruction	English

Business Storytelling

Course title	Business Storytelling
Course ID	296
ECTS	2
Course type	Elective
SWS	2
Semester	Winter and summer
Workload in hours	Total: 60 / In-class: 30 / Self-study: 30
Lecturers	Diego and Raphael Fiche
Course objectives	At the end of this course, students will be able to: • Recognize key elements that go into persuasive storytelling • Identify types of stories and their purposes • Create compelling stories to achieve business goals • Apply acquired knowledge to develop a compelling story to persuade others to think or act in a different way.
Course contents	• Introduction to Business Storytelling • Power of Business Stories: when and why to tell them • Types of Business Stories and Their Purposes • Structuring Your Story to Engage the Audience • Storytelling techniques • Enhance Your Storytelling Skills

Recommended literature	Janis Forman (2013), Storytelling in Business: The Authentic and Fluent Organization Seth Godin(2005), All Marketers Are Liars
Teaching methods	• Lectures • Group work • Case studies • Presentation • Exercises
Assessment method	Class workshops / presentation / case studies / seminar paper
Language of instruction	English
Prerequisites	None

AI Project

Course title	AI Project
ECTS	5
Course type	Project
SWS	2
Semester	Summer
Workload in hours	150 hours
Lecturer	Prof. Dr. Patrick Glauner
Course objectives	The aim of this class is to provide students with hands-on and real-world AI development experience. They will have the opportunity to work on real data sets in order to solve real-world problems. As these projects are completed in groups, students will also have the opportunity to use professional software development tools for collaboration.
Course contents	• Implementing high-tech projects in the fields of artificial intelligence, machine learning, computer vision, natural language processing and others. • Projects can be chosen for example from Kaggle, from other sources or be done in collaboration with an industrial partner. • Using modern high-end hardware, such as GPU clusters and cloud services. • Utilizing an agile process framework such as Scrum. • Understanding and using modern industrial software development tools such as work package trackers, code revision systems, debuggers, profilers and others.

	• Presenting R&D outcomes to stakeholders at different levels, such as fellow students, faculty members, practitioners and executives.
Recommended literature	1. S. Chacon and B. Straub, "**Pro Git**", Apress, second edition, 2014. 2. I. Goodfellow, Y. Bengio and A. Courville, "**Deep Learning**", MIT Press, 2016. 3. C. Larman, "**Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development**", Prentice Hall, third edition, 2004.
Teaching methods	Project
Assessment method	Project
Language of instruction	English
Prerequisite	Foundations of AI and machine learning

Algorithms and Data Structures

Course title	Algorithms and Data Structures
ECTS	5
Course type	Lecture and lab
SWS	4
Semester	Summer
Workload in hours	150 hours
Lecturer	Prof. Dr. Patrick Glauner
Course objectives	The aim of this class is to provide an introduction to one of the most important foundations of a computer science degree: algorithms and data structures. A data structure enables a programmer to structure data into conceptually manageable relationships. An algorithm is a finite sequence of well-defined, computer-implementable instructions to solve a class of problems or to perform a computation. Algorithms often operate on data structures. This course provides a journey through computer science. Students will acquire a solid foundation in how the most important algorithms and data structures work. They will also learn how to design efficient algorithms and data structures.
Course contents	• Introduction: algorithm definition, classification of algorithms • Graphs: graph definitions, applications in computer science, shortest path, lowest cost, A* • Complexity analysis: time complexity, O, Omega, Theta, o and O tilde notations, space complexity

	• Lists: arrays, dynamic arrays/lists, amortization, fundamental operations, stacks, queues, linked lists • Recursion: search, divide and conquer, recurrence relations, master theorem, backtracking, dynamic programming • Sorting: bubble sort, selection sort, insertion sort, merge sort, quicksort, lower bounds • Trees: binary trees, traversing, advanced types of trees, decision trees Maps and hash tables: key-value stores, hashing, collision handling • Selected algorithms: fast matrix multiplication, random number generation, fast inverse square root, prime numbers, Bloom filter, union-find, median of medians, string matching • Quantum computing: qubits, quantum logic gates, quantum computers, quantum algorithms
Recommended literature	1. M. Goodrich et al., "**Data Structures and Algorithms in Python**", John Wiley & Sons, 2013. 2. R. Sedgewick, "**Algorithms**", Addison Wesley, fourth edition, 2011. 3. M. Sipser, "**Introduction to the Theory of Computation**", Cengage Learning, third edition, 2012.
Teaching methods	Lecture and lab
Assessment method	Written examination 90 min.
Language of instruction	English
Prerequisite	Programming foundations

Big Data

Course title	Big Data
ECTS	2.5
Course type	Lecture and seminar
SWS	2
Semester	Summer
Workload in hours	75 hours
Lecturer	Prof. Dr. Patrick Glauner
Course objectives	The aim of this class is to provide students with an introduction to the field of big data. Students will acquire a solid foundation in how to design and implement big data systems. They will also learn hands-on how to use industrial big data tools. Furthermore, they will understand the limitations of big data-driven approaches and how they can recognize and solve typical issues in big data, such as data quality and biases. As an outcome, they will be able to work on real-world problems that not only require knowledge in AI, but also an expertise in how to use big data infrastructures, frameworks, libraries and tools.
Course contents	• Introduction: 3 Vs, history of big data, selected big data use cases • Parallelism: parallelism and concurrency, creating threads, global interpreter lock (GIL) • Big data architectures: distributed systems, MapReduce, CAP theorem, speedup through GPUs and FPGAs • Big data, small data, all data: data quality, biases in data sets, small sample size problems

	• Uncertainty in learning: confidence intervals, Gaussian processes, conformal prediction, model calibration • MLOps: project lifecycle, challenges, operations, principal components, pipelines, best practices • Big data for NLP: embeddings, recent advances in NLP, transformers • Quantum computing: qubits, quantum logic gates, quantum computers, quantum algorithms • Selected big data infrastructures, frameworks, libraries and tools
Recommended literature	1. A. Petrov, "**Database Internals: A Deep Dive into How Distributed Data Systems Work**", O'Reilly Media, 2019. 2. S. Sakr and A. Zomaya, "**Encyclopedia of Big Data Technologies**", Springer, 2019. 3. A. Tanenbaum and M. van Steen, "**Distributed Systems: Principles and Paradigms**", Pearson, 2nd edition, 2007.
Teaching methods	Lecture and seminar
Assessment method	Seminar presentation
Language of instruction	English
Prerequisite	Foundations of AI and machine learning

Computer Vision

Course title	Computer Vision
ECTS	5
Course type	Lecture and lab
SWS	4
Semester	Summer
Workload in hours	150 hours
Lecturer	Prof. Dr. Patrick Glauner
Course objectives	The aim of this class is to discuss Computer Vision (CV), which allows computers to process visual inputs. We deal every day dozens of times with CV, such as facial recognition, real-time translating camera input or auto-tagging friends in photos. Modern CV algorithms are strongly based on machine learning methods, in particular deep neural networks. Students will acquire knowledge in CV and be able to elaborate it further in the future, for example in projects or further studies. Overall, CV is a cutting-edge field, with many high-pay opportunities for graduates.
Course contents	• Introduction: applications, computational models for vision, perception and prior knowledge, levels of vision, how humans see • Pixels and filters: digital cameras, image representations, noise, filters, edge detection • Regions of images: segmentation, perceptual grouping, Gestalt theory, segmentation approaches, image compression

	• Feature detection: RANSAC, Hough transform, Harris corner detector • Object recognition: challenges, template matching, histograms, machine learning • Convolutional neural networks: neural networks, loss functions and optimization, backpropagation, convolutions and pooling, hyperparameters, AutoML, efficient training, selected architectures • Image sequence processing: motion, tracking image sequences, Kalman filter, correspondence problem, optical flow • Foundations of mobile robotics: robot motion, sensors, probabilistic robotics, particle filters, SLAM • Outreach: 3D vision, generative adversarial networks, self-supervised learning
Recommended literature	1. R. C. Gonzalez and R. Woods, "**Digital Image Processing**", Pearson, 3rd edition, 2018. 2. I. Goodfellow, Y. Bengio and A. Courville, "**Deep Learning**", MIT Press, 2016.
Teaching methods	Lecture and lab
Assessment method	Project
Language of instruction	English
Prerequisite	Programming foundations, multivariate calculus

Datacenter Network Programming

Course title	Datacenter Network Programming
ECTS	5
Course type	Lecture + Lab + Project
SWS	4
Course level	Postgraduate
Semester	Summer
Workload in hours	Total: 150 / In-class: 60 / Self-study: 90
Lecturer	Prof. Dr. Andreas Kassler
Course objectives	Students acquire understanding and hands-on experience of how the data plane of modern datacenter networking equipment can be programmed using the high-level and popular programming language P4 (see http://p4.org). They learn the basic concepts of the P4 language and understand, how offloading simple computational tasks to the data plane of programmable networking devices (such as datacenter routers or network cards) can be used to speed up the performance of Deep Learning, Big Data Analytics use-cases within modern datacenters. They understand, how the data plane can be used to accelerate distributed high-performance computing (HPC) building blocks including distributed key-value stores, where load-balancing and network monitoring of the datacenter networking fabric is important for achieving high speed and low latency. They setup their own development environment in the network emulator Mininet and implement simple data plane programs in the P4 language. They know how to use P4 to parse packet

	headers, apply different actions and modify packets before forwarding them. They know basic P4 constructs, how to store stateful information (e.g. parts of a neural network) and how to perform simple computational tasks in the data plane. Based on this knowledge and understanding, students implement a small-scale project in a team. They use their acquired knowledge on P4 and programmable datacenter networking. They evaluate the results of other project groups and get evaluated by other groups. For this project work, they have used standard tools (Mininet, P4 toolchain, command line interface) for programming the data plane of an (emulated) datacenter router. After finishing this module, students can design, implement and evaluate their own P4 programs using the network emulator Mininet.
Course contents	The Course is decomposed into two parts: Part I: "Introduction to Datacenter Network Programming" and Part II "Project in Datacenter Network Programming" Content Part I: (1) Introduction to Programming the Data Plane of a Datacenter networking device: - Difference between Data and Control Plane - Introduction to P4 language - P4 programming model - Compiling and deploying P4 programs - P4 Targets: Behavioral Model (BMv2), Programmable Switching ASIC Intel Tofino, Mellanox Bluefield DPU, Netronome SmartNIC - Basic P4 concepts: header parsing, applying tables and actions, header rewriting. - Workshop: Setup Development environment with Mininet and Command Line Interface (CLI), implement, test and debug simple P4 language constructs and programs using the Mininet network emulator (2) Datacenter Networking and Load Balancing:

	- Datacenter networking fundamentals, routing and forwarding within the datacenter networking fabric - Workshop: Advanced P4 concepts: stateful information, register arrays, counters and meters. - Loadbalancing in Datacenter networks, Equal Cost Multipath Routing, Conga, Hula - Workshop: Implementing ECMP in P4 (3) In Network support for Monitoring and Caching: - Active and passive network monitoring - Inband Network Telemetry (INT) for fine-granular network monitoring - Accelerating Distributed Key-value stores in the data plane of the data center - Using telemetry for fine-grained loadbalancing - Workshop: Implementing Hula and INT in P4 (4) In Network support for Distributed Machine Learning: - Role of the datacenter network for distributed training and inference - In network support for Distributed Machine Learning Inference for in-switch traffic classification - Mapping trained machine learning models (decision trees, SVMs, neural networks) to programmable data plane devices - In network support for distributed training within a datacenter network Content Part II: Project: Implementation of your own small dataplane program in P4 and testing it in the Mininet network emulator.
Recommended literature	Recommended Literature will be provided at the start of the course by a set of research and practical oriented articles that are available online.
Teaching methods	Lecture with exercises and Labs followed by a small scale project

Assessment method	Written examination, 90 min.
Language of instruction	English
Prerequisites	Students should have basic understanding of Network Technologies and/or Communication Networks. Basic knowledge of Programming and basic knowledge in Python helps in the Project Part of the course.

Database Engineering

Course title	Database Engineering
ECTS	5
SWS	4
Course type	Undergraduate
Semester	Winter and summer
Workload in hours	In-class: 60 hrs. / Self-study: 90 hrs / Total: 150 hrs
Lecturer	Dr. Michael Scholz
Course objectives	After this module students should • be able to describe the database design process, • know the elements of the Entity-Relationship-Model, • can build an Entity Relationship Model for a specific case, • can normalize a database design, • be able to manage a database through a database management system, • be able to query a database using SQL, • know the core components and functionalities of a database management system.
Recommended literature	Conolly, Thomas M.; Begg, Carolyn E.: Database Solutions - A step-by-step guide to building databases. 2nd Edition. Harlow, Essex: Pearson Education Limited, 2004 Conolly, Thomas M.; Begg, Carolyn E.: Database systems - A practical approach to design, implementation, and management. 4th edition. Addison-Wesley, an imprint of Pearson Education, 2005

Teaching methods	Classes with exercises and practical training Course and document management through E-Learning System iLearn
Assessment method	Written examination, 90 min.
Language of Instruction	English
Prerequisites	Basics in Computer Science

Advanced Programming Techniques

Course title	Advanced Programming Techniques
ECTS	5
Course type	Lecture
SWS	4 (2 SWS Lecture + 2 SWS Lab)
Course level	Postgraduate
Semester	Summer
Workload in hours	Total: 150 / In-class: 60 / Self-study: 90
Lecturer	N.N.
Course objectives	The students advance their knowledge in computer programming with the goal to create and maintain complex software applications. By regular lab sessions, the students learn the interplay between design, implementation, operation and evolution of modern software in a hands-on manner.
Course contents	The Python Programming Language, Version Control Systems, Advanced Software Engineering Processes, UML Modeling, Software Design Patterns, Unit Testing, Defensive Programming, User Interface Design

Recommended literature	[1] Michael Goordrich, Roberto Tamassia, Michael Goldwasser. Data Structures and Algorithms in Python. 1st ed. 2013, John Whiley & Sons. [2] Robert C. Martin: Clean Code, *A Handbook of Agile Software Craftmanship*. 1st ed. 2008, Prentice Hall [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: *Design Patterns. Elements of Reusable Object-Oriented Software*. 1st ed. 1994, Prentice Hall [4] Ian Sommerville, Engineering Software Products: An introduction to modern Software Engineering. 1st Edition. 2020, Person Education.
Teaching methods	Lecture with lab sessions
Assessment method	Written examination, 90 min
Language of instruction	English
Prerequisites	Basic knowledge in object-oriented programming and operating systems.

Advanced Topics in AI

Course title	Advanced Topics in AI
ECTS	5
Course type	Lecture
SWS	4
Course level	Postgraduate
Semester	Summer
Workload in hours	Time of attendance: 60 hours Self-study: 90 hours Total: 150 hours
Lecturer	Prof. Dr. Andreas Fischer
Course objectives	The purpose of this course is to provide students with hands-on and real-world development experience. They will have the opportunity to review some cutting-edge research papers and to then turn them in concrete software/hardware outcomes. As these projects are completed in teams, students will also have the opportunity to elaborate on their social and language skills. At the end of the term, students will present their projects at an in-house R&D fair which will be open to the public.
Course contents	– Implementing contemporary research papers from the fields of artificial intelligence, machine learning, computer vision, natural language processing and others. – Using modern high-end hardware, such as GPUs clusters and cloud services. – Utilizing an agile process framework such as Scrum.

	– Understanding and using modern industrial software development tools such as work package trackers, code revision systems, debuggers, profilers and others. – Presenting R&D outcomes to stakeholders at different levels, such as fellow students, faculty members and practitioners and executives.
Recommended literature	Basic: - C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006. - I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016. Study aids: - High-end GPUs - Cloud services - Development boards - Mobile robots and drones - Hardware manuals
Teaching methods	project and seminars
Assessment method	written student research project
Language of instruction	English
Prerequisites	None

ChatGPT et al.: Generative AI with Transformers

Course title	ChatGPT et al.: Generative AI with Transformers
ECTS	5
Course type	Lecture
SWS	4
Semester	Summer
Workload in hours	Total: 150 / In-class: 60 / Self-study: 90
Lecturer	Prof. Dr. Andreas Fischer
Course objectives	The module will give an introduction to the transformer technology which drives modern large language models, such as ChatGPT.
Course contents	Covered topics are in particular: • Foundations of Language Models • Word Embeddings • Attention Mechanism • Architectures of Transformer Models • Popular Open Source Transformer Models • Limitations of Large Language Models • Applications of Transformers in and beyond NLP • Optimization of Transformer Models
Recommended literature	tba

Teaching methods	Lecture
Assessment method	Project work (paper)
Language of instruction	English
Prerequisites	None

Theoretical Foundations of AI and Data Science

Course title	Theoretical Foundations of Artificial Intelligence and Data Science
ECTS	8
Course type	Lecture
SWS	6
Semester	Summer
Workload in hours	Total: 240 / In-class: 90 / Self-study: 150
Lecturer	Prof. Dr.-Ing. Markus Mayer, Prof. Dr. Peter Faber
Course objectives	In this module, students become familiar with the theoretical foundations of computer science such as: • Formal logic, • Probability, • Machine models (with applications in complexity theory and predictability theory), • Formal languages and their hierarchies. This course lays the foundation for understanding the theoretical background for algorithms and methods inside and outside of Data Science and AI applications: Early AI systems and prominent AI examples are based on formal logic. A majority of Data Science methods directly originates from Statistics, which in turn is based on probability. This field (probability) also builds the groundwork for the most prominent modern AI and Data Science method - Machine learning. In addition, the students gain an understanding of complexity estimates and verification options, as well as the basic functions of data processing systems.

Course contents	Main topics: **Probability:** • Counting and sets • Probability experiments, Toy examples (coin, dice, urn) • Conditional probability, independence and Bayes theorem • Discrete and continuous random variables • Expected value, standard deviation and variance • Central limit theorem and the Law of large numbers • Joint distributions and independence • Covariance and correlation **Formal logic:** • Boolean algebra • First-order logic • Logic proofs **Formal languages and compiler construction:** This course illuminates the theoretical background behind formal languages and language processing: Chomsky hierarchy, giving a detailled description of (non-)deterministic finite automata, regular languages, push-down automata, context-free languages, Turing machines, recursively enumerable languages, computability (halting problem), and their applications in compiler construction -- lexer, parser, compiler frontend. • Introduction and Translators I • Translators II / Formal Languages I • Formal languages II / III • Lexical Analysis I / II • Syntactic analysis I / II • Syntactic analysis III / IV • Syntax directed translation I / II • Summary, possibly further topics (e.g. on optimizing compiler construction) - wrap-up, possibly further topics (e.g., optimizing compilers)

Recommended literature	Probability: The course uses the script of the MIT Open CourseWare course "Introduction To Probability And Statistics", https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2014/. The script was developed by Dr. Jeremy Orloff and Jonathan Bloom and can be downloaded under a CC license. Formal logic: • John Heil (2021), First-Order Logic: A Concise Introduction, Hackett Publishing Co, Indianapolis, USA • Russell, S., Norvig, P. (2016), Artificial Intelligence - A Modern Approach, Fourth Edition. Pearson, London, GB • Eric Steinhart (2027), More Precisely: The Math You Need to Do Philosophy, Broadview Press Ltd, Peterborough, Canada Basic literature (Formal languages and compile building): • John Longley, Lessons in Formal Programming Language Semantics, University of Edinburgh, 2003 • F.L. Bauer, H. Wössner: Algorithmische Sprache und Programmentwicklung, Springer Verlag 1984 • (availble also in English) • Rudolf Berghammer: Semantik von Programmiersprachen, Logos Verlag, 2001 • Juraj Hromkovic: Theoretische Informatik, Springer Verlag • Uwe Schöning: Theoretische Informatik - kurz gefasst. Spektrum, 2008 • Hopcroft, Motwani, Ullman: Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 2001 • Hopcroft, Motwani, Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson, 2002. • Compilers Principles, Techniques, and Tools; Aho, Lam, Sethi, Ullmann; 2nd edition; Addison-Wesley; 2007 • Engineering a compiler; Cooper, Torczon; 2nd Edition, Morgan Kaufmann 2012 • Introduction to Automata Theory, Languages, and Comoputation; Hopcroft, Motwani, Ullman; Addison-Wesley; 2001

Teaching methods	Lecture with exercises
Assessment method	Written examination, 90 min.
Language of instruction	English
Prerequisites	• Programming in an advanced programming language (e.g. C, C++, Java, C#) • Mathematics of natural numbers (induction) • Analysis, Linear Algebra

Machine Learning

Course title	Machine Learning
ECTS	5
Course type	Lecture / Exercises / Project
SWS	4
Semester	Summer
Workload in hours	Total: 150 / In-class: 60 / Self-study and project work: 90
Lecturer	Prof. Dr.-Ing. Markus Mayer
Course objectives	• The students understand how to analyse a given dataset for its predictive quality and know how to motivate a classification or regression problem. They can perform the computation of statistics and programming of visualizations and can select the appropriate methods for given datasets. • The students can present a problem of supervised learning with remarks on predictive quality and the motivation. • The students understand the basic methods of feature engineering and the construction of machine learning evaluations in a scientific, rigid way. They can apply this knowledge and implement the methods in code. • The students understand the basic classification methods and their advantages and disadvantages. For a given problem, they can decide which to include in an evaluation and appropriatly parameterize library methods. • The students know unsupervised learning and some of its usages. They understand an exemplary algorithm from this field and can do the implementation.

Course contents	• Machine Learning Introduction • Machine Learning Overview: Exemplary problems and solution classes • Supervised learning experiment: Motivation • Error and quality measures • Bayes classifier, kNN classifier • Training and test data selection • Bayesian type classifiers • Linear regression • Feature engineering (Creation, Lifting, Selection) • Outliers, Cross validation, Resampling • Unsupervised Learning, K-Means clustering • Gradient descent • Support vector machines • Decision tree classifiers The main body of the lecture is supervised learning. Undsupervised learning is motivated and only exemplary shown.
Recommended literature	• An Introduction to Statistical Learning, James, Witten, Hastie, Rob Tishirani, 2nd Edition, 2021, available online: https://www.statlearning.com • MIT Open learning library: Introduction to machine learning. Online course, available at https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.036+1T2019/about
Teaching methods	• Lecture with PowerPoint slides • Exercises in the lecture and for self study • Presentation of the exercises by students with discussion • Online course material for self study • Project work with mandatory lecturer meetings as exam
Assessment method	Project work, ongoing over the semester
Language of instruction	English

Prerequisites	Analysis, Linear Algebra, Statistics

Project Management

Course title	Project Management
ECTS	5
Course type	Lecture
SWS	4
Semester	Summer
Workload in hours	Total: 150 / In-class: 60 / Self-study: 90
Lecturer	Prof. Dr. Christina Bauer
Course objectives	The students get to know the most important content of (IT) project management. After the course the students are able to plan and carry out a project with appropriate methods.
Course contents	Contents include but are not limited to: ▪ Phases of a project and documentation ▪ Requirements engineering ▪ Project controlling ▪ Static and agile methods
Recommended literature	▪ Cleland, D. I., & Ireland, L. R. (2008). Project manager's handbook: Apply best practices across Global industries. McGraw-Hill. ▪ Additional literature will be announced in the course
Teaching methods	Combination of lecture, presentation and case studies
Assessment method	Written examination, 90 min. and presentation

Language of instruction	English
Prerequisite	none