DEGGENDORF INSTITUTE OF TECHNOLOGY

Table of Contents

German A1/ Parts 3 and 4

Course title	German A1/ Parts 3 and 4
ECTS	4
Course type	Course with exercises
SWS	4
Semester	Winter and Summer
Workload in hours	60 hrs
Lecturer	Dr. Virginia Wallner
Course objectives	• Can understand and use familiar expressions and very basic phrases aimed at meeting concrete everyday needs • Can introduce themselves and others and ask other people questions about their person • Can communicate in a simple way if the other person speaks slowly and clearly and is willing to help http://www.europaeischer-referenzrahmen.de
Course contents	• Grammar – Prepositions – Possessives – Dative verbs – The imperative-Simple past 'war/ hatte' – The perfect form – Word formation – Subjunctive II • Topics – Apartments and houses – Parts of the body – Describing people and their character – Household activities – Weather – Holidays and celebrations

Recommended literature	Menschen. Deutsch als Fremdsprache. Kursbuch A1.2 Hueber. Kapitel 13-24 ISBN 978-3-19-561901-1 Menschen. Deutsch als Fremdsprache. Arbeitsbuch A1.2 mit Audio-CD. Hueber. Kapitel 13-24 ISBN 978-3-19-511901-6
Teaching methods	• Partner and group work • Explanation of topics by the lecturer • Presentations and discussions • Feedback from the lecturer • Listening exercises
Assessment method	Written examination, 90 min.
Language of instruction	German
Prerequisites	Successful completion of Level A1/Parts 1 and 2 (88121)

Course descriptions for German language courses at higher 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 http://www.europaeischer-referenzrahmen.de/
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.
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

Basics of International Sales and Business Development

Course title	Basics of International Sales and Business Development
Course ID	268
ECTS	2
Course type	Lecture with group work and presentations
SWS	2
Semester	Winter and summer
Lecturer	Ibrahim Waked
Course objectives	General knowledge of international sales and strategic business development mechanisms. As well as profound analysis of practical case studies.
Course contents	• Basics of sales and business development • Analysis of market potential including cultural & political aspects, correlation between microeconomic and demographic aspects, (PESTELO analysis) • Relevancy of world bank reports on general economic performance and their implementation in company BD strategy • Market entry and risk management
Recommended literature	**Strategic Management** by Richard Lynch von Pearson Longman **Business Development Management** By Lutz Becker, Walter Gora, Tino Michalski
Teaching methods	Lecture with integrated project development examples
Assessment method	Presentation and seminar paper
Language of instruction	English

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 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

Cross-Cultural Team Building

Course title	Cross-Cultural Team Building Workshop
Lecturer	Prof. Dr. Johann Nagengast
Course type	Elective
SWS	2
Semester	Winter and summer
ECTS	2
Assessment method	Seminar paper
Course language	English
Course objectives	Globalisation demands that managers possess the basic skills required to work together in international teams. Many companies actively encourage the development of these skills through teambuilding or team development programs. Especially for change management, team development plays an increasingly important role. Here the critical goal is to optimise how the group members work together as a team. Key factors affecting a team's success include organisation, structures, processes, culture and relationships. International Team Building is conducted at the beginning of the semester as a three day off-campus seminar. The hands-on, outdoor training gives the students intensive exposure to the multifaceted nature of group dynamics. By working together to solve complex problems and through structured feedback sessions, the participants become sensitised to the rolls they assume in group interactions, to the limitations imposed by the German and their own cultures, and to the conditions required for effective team work.

	The course supports the integration of foreign students into campus and social life and helps build lasting working relationships among all participants. The skills of giving and receiving of feedback are learned in the protective atmosphere of small groups through intensive exchanges between instructors and participants. This leads to improved observation and communication skills. Moreover, the group members continually switch roles. This promotes a deeper understanding of social interaction, helps members to reflect on their contribution to the group process, encourages members to experiment with new behavioural concepts, and improves the group's capacity to co-operate and perform. Final feedback rounds offer the possibility to align the members' self-images with the perception others have of them, to reduce "blind spots", to increase self-confidence and their ability to reflect. The capacity to give appropriate feedback in various situations, to monitor one's self-image as well as the consequences of one's own behaviour form the basis for a successful career in management.
Course contents	Group dynamics, processes and structures in groups; Roles in groups (roles in tasks and supporting roles); Group leadership; Effect of one's actions in groups; The "give and take" of feedback; Self-image and how others see you; Communication levels (content versus relationship); Conditions for successful co-operation; Cultural influences on teamwork. Note: The main emphasis of this course is not the conveyance of theoretical knowledge, but rather learning directly from experience. The theories on which the intervention and evaluation sessions are based are taught in the course "Human Resources Management".
Teaching methods	This course is organised as an interactive experience and activity-based training program. With the help of complex tasks, timed interaction activities combined with elements of surprise, classical outdoor training exercises, moderated feedback and reflection sessions, participants are taught the necessary conditions for effective teamwork. The teaching methods are based on the principles of self-organised learning. The instructors define their roles in terms of Schein's model of process consulting.

	They intervene by questioning the participants in a manner designed not only to examine their perspectives, but to introduce new perspectives and stimulate the group's creative process. The responsibility for these process remains with the participants. In the context of the learning environment, the students enjoy the opportunity to increase their observation, communication, co-operation, self-reflection, teamwork and management skills as well as their self-confidence. In addition, the course offers the students the chance to network and develop sustainable work relationships at the start of their studies.
Suggested Literature	Baron, R. S.: Group Process, Group Decision, Group Action, 2nd. Ed., Buckingham, 2003; Buchanan, D., Huczynski, A.: Organizational Behavior, 5th Ed., Harlow, 2004; Wagner, M., Waldmann, R.: Vom Outdoor-Training zur Teamentwicklung, Welchen Beitrag leisten Hochseilgärten? in: Jagenlauf, M./Michl, W. (Hrsg.) Erleben und Lernen – Internationale Zeitschrift für handlungsorientiertes Lernen, 1/2004
Notes	The weekend seminar is characterised by team teaching in a mountain hostel. The team consists of Prof. Dr. Nagengast and trained tutors selected from participants in the course „Train the Trainer". The tutors make it possible to conduct the training in small „protected" groups (around 8) and to give qualified feedback.

AI Project

Course title	AI Project
ECTS	5
Course type	Project
SWS	2
Course level	Undergraduate
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
Course level	Undergraduate
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
Course level	Undergraduate
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) • Revision of database fundamentals: ER diagrams, relational databases, database management systems, queries, indexes, normalization, transactions • 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

	• 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
Course level	Undergraduate
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	Undergraduate
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.

Databases

Course title	Databases
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	Prof. Dr. Wolfgang Dorner / Prof. Dr.-Ing. Udo Garmann
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	Prof. Dr. Andreas Wölfl
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

Informatics II

Course title	Informatics II - Contemporary Usage of Python
ECTS	5
Course type	Lecture
SWS	4
Course level	Postgraduate
Semester	Summer
Workload in hours	Time of attendance: 60 hours self-study: 45 hours virtual learning: 45 hours Total: 150 hours
Lecturer	Prof. Dr. Gökçe Aydos
Course objectives	The purpose of the course is to learn to: **Professional competences:** • outline fundamental features of the Python programming language • understand the advantages of object-oriented and functional programming • know different request types to access web resources • list useful libraries from the standard library **Methodological competences:** • implement programs for string processing • leverage the interactive interpreter for short computing tasks • use object-oriented programming to breakdown a program into classes • use functional programming to write shorter code • implement programs for interacting with web APIs • carry out simple image processing tasks

	• leverage Numpy to conveniently work with matrices • use an unknown library by reading its documentation **Social competences** • cooperate in a pair programming setting • evaluate someone else's work and give constructive feedback (e.g., in context of peer-assessed exercises)

Course contents	Most of the contents are based on the course CS41: The Python Programming Language from Stanford University.
	
	• Python basics:
	  ○ Interactive interpreter
	  ○ Comments
	  ○ Variables and types
	  ○ Numbers and Booleans
	  ○ Strings and lists
	  ○ Console I/O
	  ○ Control Flow
	  ○ Loops
	  ○ Functions
	  ○ Assignment Expressions
	• Data structures:
	  ○ list
	  ○ dict
	  ○ tuple
	  ○ set
	• Object-oriented Python:
	  ○ errors and exceptions
	  ▪ *easier to ask for forgiveness than permission* (EAFP) vs *look before you leap* (LBYL)
	  ○ data model
	  ○ classes
	  ○ exceptions as classes
	• Functions:
	  ○ namespaces and scope
	  ○ Python Functions
	  ○ (variadic) arguments
	  ○ Parameter ordering
	• Functional programming:
	  ○ meaning
	  ○ first-class functions
	  ○ lambdas
	  ○ iterators and generators
	  ○ map and filter
	  ○ decorators

	• Python & the Web:
	  ○ HTTP
	  ○ requests library
	  ○ working with images
	  ○ creating a web interface for your app using Flask library
	• Numpy:
	  ○ what is a matrix?
	  ○ why are matrices useful?
	  ○ n-dimensional array ndarray
	  ○ axes and shapes
	  ○ matrix operations
	  ○ statistical methods
	  ○ parameter fitting example
	• Standard library and third-party libraries

Recommended literature	• Lecture videos in DIT's Moodle
	  ○ alternatively: Stanfordpython course reader
	• Slides that accompany the videos. These are tailored for discussions during the class.
	• Exercise notebooks in DIT's Moodle
	• Previous Exams
	  ○ Instructor versions
	  ○ Student versions

Teaching methods	• Flipped classroom • Labs with feedback sessions • Pair programming
Assessment method	Written examination, 90 min.
Language of instruction	English
Prerequisites	• Computer science fundamentals (e.g., information, hardware, software, operating systems, shells, algorithms) • Fundamental programming tools (e.g, control flow, data structures, functions)

FPGA Programming

Course title	FPGA Programming - Digital Circuit Design Using Systemverilog
ECTS	5
Course type	Lecture
SWS	4
Semester	Summer
Workload in hours	Total: 150 / In-class: 60 / Self-study: 90
Lecturer	Prof. Dr. Prof. Dr.-Ing. Gökçe Aydos
Course objectives	The purpose of the course is to learn to: • explain the typical structure of FPGAs • differentiate a hardware description language from a typical programming language, e.g., regarding structure and purpose • use Systemverilog and a state-of-the-art FPGA development tool to develop circuits on an FPGA • differentiate between structural and behavioral design approaches • analyze behavioral description and write code that implement the behavior • classify which data processing tasks are better suited for FPGAs than general processors • cooperate in a pair programming setting • evaluate someone else's work and give constructive feedback (e.g., in context of peer-assessed exercises)

Course contents	• Getting started with the FPGA board
	• Implementation (including testing using testbenches) of:
	  ○ combinational logic, e.g., multiplexer, decoder, shifter, encoder
	  ○ sequential logic, e.g., flip-flop, latch, counter, memory
	  ○ arithmetic circuits, e.g., adder, multiplier

	○ state machines ○ a digital system: reaction time monitor
Recommended literature	• Digital logic course by Realdigital ○ Lecture videos related to the content can be found in the section *Lectures* on EE214 course page from Washington State University • Accompanying lecture notes: Digital logic notes • Realdigital Boolean FPGA board will be provided by the instructor that you may use during the class. It is not possible to borrow a board for home. If you want to work at home, please buy one.
Teaching methods	To reach the learning outcomes we will use the following didactic methods: • Flipped classroom • Labs with feedback sessions and pair programming During the labs you are encouraged to work with a partner in a pair programming setting. Your partner and the instructor will give you feedback. • Mini projects on the FPGA Every week there will be problems that you must solve on an FPGA board.
Assessment method	Written examination, 90 min.
Language of instruction	English
Prerequisites	• Fundamental programming tools (e.g, control flow, data structures, functions) • Digital logic (e.g., transistor, logic gate, K-map, SOP, POS, multiplexer, counter) The learning materials contain a graceful introduction to digital logic so you can still attend the course if you do not have any experience with digital logic. But expect more workload in this case.